Natural Image Statistics for Digital Image Forensics

A Thesis Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Computer Science

by

Siwei Lyu

DARTMOUTH COLLEGE
Hanover, New Hampshire
August, 2005

Examining Committee:	Hany Farid, Ph.D. (chair)
	George V. Cybenko, Ph.D.
	Daniel Rockmore, Ph.D.
	William T. Freeman, Ph.D.
Dean of Graduate Studies:	Charles K. Barlowe, Ph.D.

Dartmouth College
Department of Computer Science
Technical Report TR2005-557

This thesis is dedicated to my wife, Yanfei Chen,
and to my parents, Changmin Lyu and Jingfan Wu,
for their endless love and support.

Abstract

Natural Image Statistics for Digital Image Forensics

Siwei Lyu

We describe a set of natural image statistics that are built upon two multi-scale image decompositions, the quadrature mirror filter pyramid decomposition and the local angular harmonic decomposition. These image statistics consist of first- and higher-order statistics that capture certain statistical regularities of natural images. We propose to apply these image statistics, together with classification techniques, to three problems in digital image forensics: (1) differentiating photographic images from computer-generated photorealistic images, (2) generic steganalysis; (3) rebroadcast image detection. We also apply these image statistics to the traditional art authentication for forgery detection and identification of artists in an art work. For each application we show the effectiveness of these image statistics and analyze their sensitivity and robustness.

Acknowledgments

First and foremost I would like to thank my advisor, Dr. Hany Farid, for his advice, guidance and support over the years, to which work described in this thesis is attributed.

I would also like to thank all my professors at Dartmouth College, and my wonderful colleagues and friends. In particular, I would like to thank:

My thesis committee members: Dr. George Cybenko, Dr. Bill Freeman, and Dr. Dan Rockmore,

Former and current members from the Image Science Group: Kimo Johnson, Dr. Alin Popescu, Dr. Senthil Periaswamy, and Weihong Wang,

My professors at Dartmouth: Tom Cormen, Devin Balcom, Lorie Loeb, Prasad Jayanti, and Bruce Donald,

The Sudikoff Lab system administrators: Sandy Brash, Wayne Cripps, and Tim Trebugov, and department administrators, Kelly Clark, Sammie Travis and Emily Holt Foerst,

My friends and colleagues: Dr. Meiyuan Zhao, Fei Xiong, Dr. Lincong Wang, and Anthony Yan.

Contents

Chapter 1

Introduction

Our perceptual systems have evolved to adapt and operate in the natural environment. Among all types of sensory data that probe the surrounding environment, images provide the most comprehensive information critical to survival. Accordingly, human vision is most sensitive to images that resemble scenes in the natural environment. Such images are usually called natural images[1].

However, among the set of all possible images, natural images only occupy a tiny subspace [16, 36, 14, 62]. This is more easily seen in the digital domain. For instance, there are totally 256^{n^2} different 8-bit grayscale images of size $n \times n$ pixels (with as few as $n = 10$ pixels, it results in a whopping 1.3×10^{154} different images). Considering multiple color channel images makes the number more striking. Yet if we uniformly sample from this colossal image space by randomly choosing one grayscale value from $\{0, 1, \cdots, 255\}$ for each pixel, most of the time a noise pattern lack of any consistent structure will appear, and with a very rare chance it will produce a natural image, Figure 1.1. The fact that such a uniform sampling seldom produces a natural image confirms that natural images are sparsely distributed in the space of all possible images.

Although relatively small in number, natural images exhibit regularities that distinguish themselves from the sea of all possible images. For example, natural images are not simply a collection of independent pixels. The visual structures making them look "natural" are the result of strong correlations among pixels [62]. The regularities within natural images can be modeled statistically, and have had important applications in image compression [6, 43, 81], denoising [49, 58], segmentation [29], texture synthesis [85, 28, 57], content-based retrieval [4] and object/scene categorization [74]. In general, the statistical descriptions of natural images play three important roles: (a) prior or regularization terms for probabilistic inference about natural images; (b) constraints for sampling procedures for special class of natural images, such as textures; (c) discriminative/descriptive features for differentiating or describing natural images. For computational efficiency, most existing statistical descriptions of natural images take some simple mathematical forms, such as individual statistics [52, 16, 62], parametric densities [45, 67], and Markov random fields [19, 12, 25].

Statistical descriptions of natural images also have important applications in the burgeoning field of digital image forensics. The popularity of digital images makes it also vulnerable to tam-

[1] Rigorously defining the "naturalness" of an image is very difficult as the term is highly subjective.

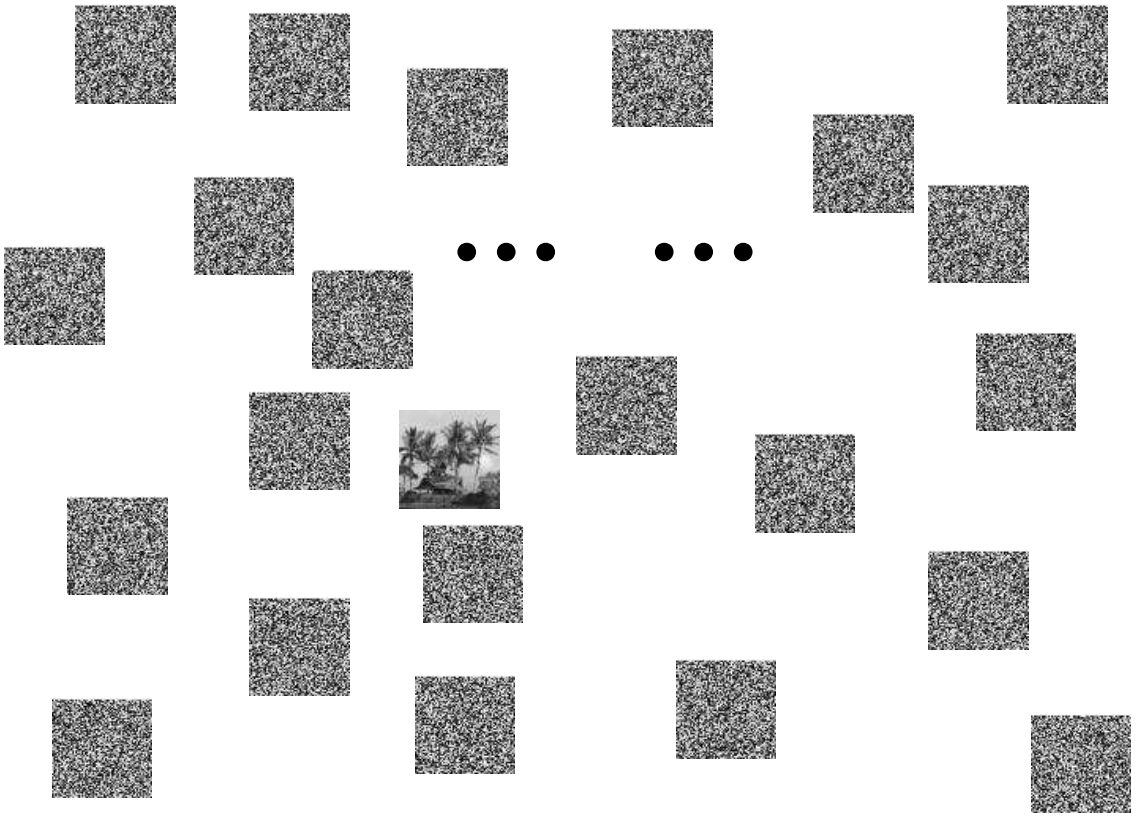


Figure 1.1: Natural images are rare in all possible images, but they are not random.

pering. An image not only tells a story, but may also tell a secret. An innocent-looking natural image may conceal a hidden message or generated by a computer graphics program. Such manipulations, along with tampering with image contents [54, 55], have become increasingly easy as the development of technologies. Digital image forensics are techniques aiming to shed lights on such secrets beneath digital images, and is therefore of the great interest to the law-enforcement and national security agencies.

Contributions

Specifically, we are interested in the following problems in digital image forensics:

Photographic or photorealistic: Sophisticated computer graphics softwares can generate highly convincing photorealistic images able to deceive the human eyes. Differentiating between these two types of images is an important task to ensure the authenticity and integrity of photographs.

Generic image steganalysis: Image steganography hides messages in digital images in a non-intrusive way that is hard to detect visually. The task of generic steganalysis

is to detect the presence of such hidden messages without the detailed knowledge of the embedding methods. Because of the potential use of steganography as a covert communication method, steganalysis is of interest to the law-enforcement, counter-intelligence and anti-terrorism agencies.

Live or rebroadcast: Biometrics-based (e.g., face, iris, or voice) authentication and identification systems are vulnerable to the "rebroadcast" attacks. An example is the defeat of a face recognition system using a high-resolution photograph of a human face. An effective protection against such an attack is to differentiate a "live" image (captured in real time by a camera) and a "rebroadcast" one (a photograph).

Common to these problems is the task of differentiating natural photographic images from some other classes of images: in photographic vs. photorealistic, the other class is the computer-generated photorealistic images; in steganalysis, they are images with steganographic messages; and in live vs. rebroadcast, they are the prints of natural images. Instead of seeking an individual solution to each problem, we propose to solve them in a unified framework. First, statistics capturing certain aspects of natural images are collected. Using these statistics to form discriminative features, classification systems are built to determine the class of an unknown image. At the core of this image classification framework is a set of image statistics as discriminative features. Over the years, many different image features characterizing different aspects of image contents have been proposed in various context [4, 27, 73]. However, these features are less suitable to the digital image forensics application, where difference in image contents is irrelevant. Very few image features are designed for characterizing natural images as a whole ensemble.

In this work, we propose a new set of image statistics based on observations of natural images and demonstrate their applications in digital image forensics. The proposed image statistics are collected from multi-scale image decompositions and are empirically shown to capture certain fundamental properties of a natural image. We then apply these image statistics, together with non-linear classification techniques to solve the three digital image forensics problems, and empirically demonstrate their effectiveness. Furthermore, these image statistics are extended to help the traditional field of art authentication, where they also show promising performance. We also evaluate the overall robustness and sensitivity of these image statistics in face of some common image manipulations.

The rest of the thesis is organized as following: in Chapter 2, the image statistics are described in detail, and experiments on natural and synthesized images are presented to justify their uses. In Chapter 3, techniques to build effective classifiers are introduced. This is followed by the sensitivity analysis of the proposed image statistics and classification systems in Chapter 4. Chapter 5 focuses on the application of the image statistics and classification techniques to the problem of differentiating photographic and computer generated photorealistic images. Chapter 6 is on their application to generic image steganalysis. In Section 7.1, the problem of live vs. rebroadcast is solved in a similar framework. In Section 7.2, we extrapolate the application of a variant of this set of image statistics to art authentication. Chapter 8 concludes the thesis with a general discussion.

Chapter 2

Image Statistics

In this chapter we describe in detail the image statistics proposed in this work. These statistics are collected from image representations that decompose an image using basis functions that are localized in both spatial positions and scales, implemented as a multi-scale image decomposition. In Section 2.1, two multi-scale image decompositions, namely, the quadrature mirror filter (QMF) pyramid decomposition and the local angular harmonic decomposition (LAHD) are described. Image statistics collected from these two decompositions are described in Section 2.2. In Section 2.4, experiments are performed to show that these statistics capture some non-trivial statistical regularities in natural images.

2.1 Image Representation

At the core of any statistical description of natural images is the choice of a suitable image representation. There are, of course, many different image representations to choose from. The choice should be made based on their effectiveness in revealing statistical regularities in natural images. The simplest representation, for example, is an intensity-based approach, where the representation is simply the original intensity values. An $n \times n$ grayscale image is considered as a collection of n^2 independent samples of intensity values. Similarly, an $n \times n$ RGB color image is represented as a collection of n^2 independent 3D vectors. The most standard statistical description of an image for the intensity-based representation is the histogram of intensity values, which gives the distribution of intensity values of the image. Shown in the first two panels in the top row of Figure 2.1 are two images with exactly the same intensity histograms (third panel). However, these two images are quite different visually, with the left one being a natural image and right one being a noise pattern lacking coherent structures. It is generally difficult to capture the statistical regularities of natural images in the intensity domain, as suggested by this example.

Another popular image representation is that based on a global Fourier decomposition. In this representation, an image is first decomposed as a sum of sines and cosines of varying frequency and orientation: $F(\omega_x, \omega_y) = \sum_x \sum_y I(x, y)\cos(\omega_x x + \omega_y y) + \imath \sin(\omega_x x + \omega_y y)$, where $I(x, y)$ is a grayscale image, and $F(\omega_x, \omega_y)$ is its Fourier transform (each channel of a color image is independently represented in the same way). The spatial frequency of the sine/cosine is given

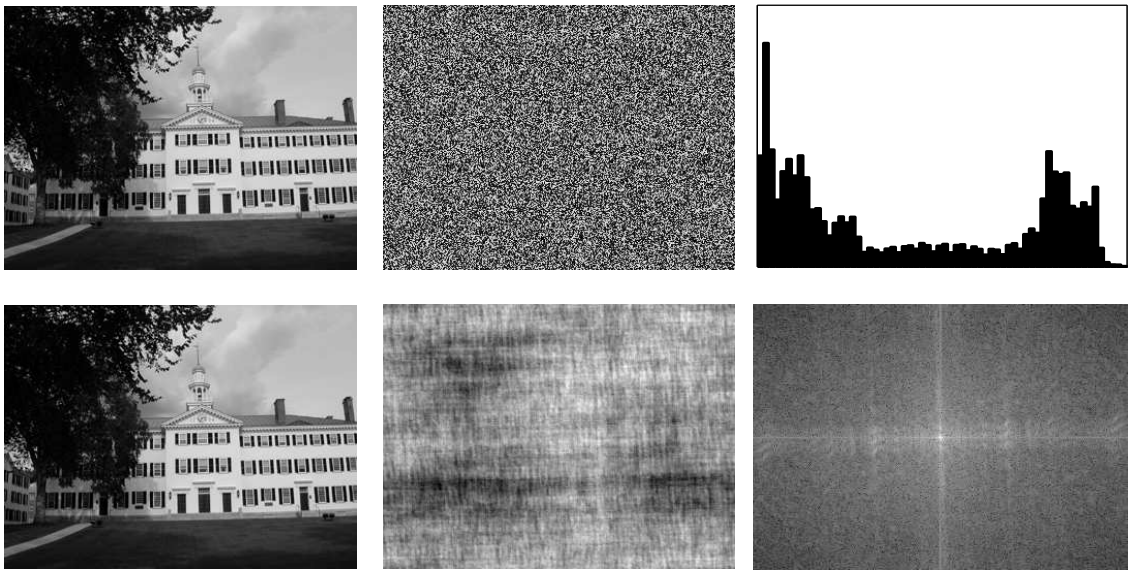


Figure 2.1: Shown in the top row are a pair of images with identical intensity histograms (third panel). Shown in the bottom row are a pair of images with identical Fourier magnitudes (third panel).

by (ω_x, ω_y) and their orientations are measured by $\tan^{-1}(\omega_x/\omega_y)$. It has been observed that the magnitudes of each frequency component, $|F(\omega_x, \omega_y)|$ of natural images are often well modeled with an exponential fall-off: $|F(\omega_x, \omega_y)| \propto (\omega_x^2 + \omega_y^2)^{-p}$, where the exponent p determines the rate of the fall-off [65]. Such a regularity reflects the scale invariance in natural images. However, this is not sufficient to characterize a natural image, as shown in the first two panels in the bottom row of Figure 2.1 as two images with exactly the same Fourier magnitude (third panel). Such a Fourier-based representation is not sufficiently powerful to discriminate between an image and a "fractal-like" pattern, which suggests that a global Fourier representation is not particularly useful in capturing statistical regularities in natural images.

The intensity- and Fourier-based representations are, in some ways, at opposite ends of a spectrum of representations. The basis functions for the pixel-based representation are perfectly localized in space, but are infinite in terms of their frequency coverage. On the other hand, the basis functions for a Fourier-based representation are perfectly localized in frequency, but are infinite in the spatial domain. Image representations based on multi-scale image decomposition (e.g., wavelets) decompose an image with basis functions partially localized in both space and frequency [70], and thus offer a compromise between these representations, Figure 2.2. As natural images are characterized by spatial-varying localized structures such as edges, these representations are generally better than intensity- or Fourier-based representations at describing natural images. Within the general framework of multi-scale image decomposition, there exist many different implementations, each having its own advantage and effective in different problems. In this work, two such decompositions, namely the quadrature mirror filter (QMF) pyramid decomposi-

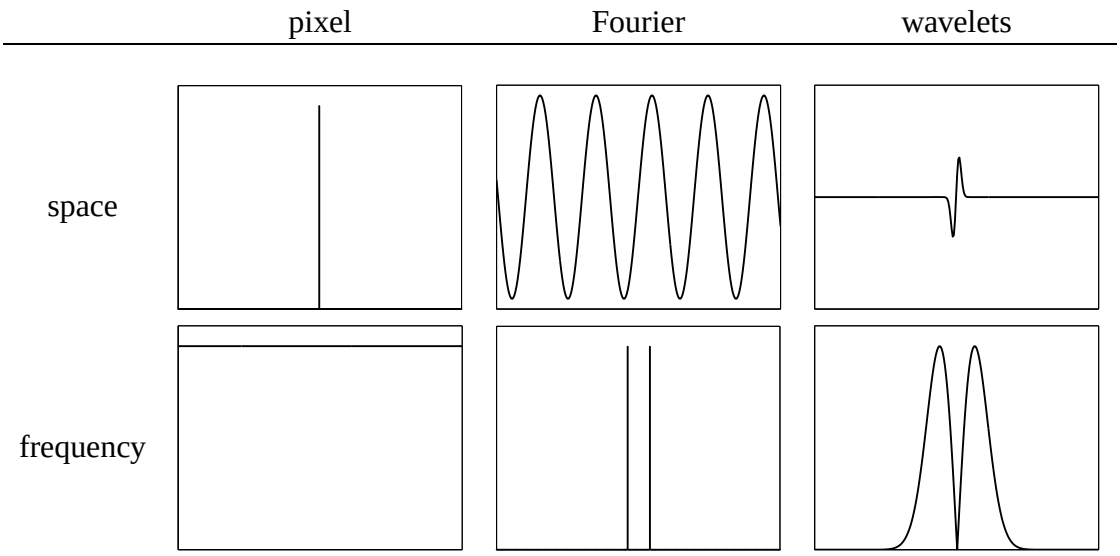


Figure 2.2: Shown are 1-D space and frequency (magnitude) representations of pixel, Fourier, and wavelet-like basis functions.

tion and the local harmonic angular decomposition (LAHD), are employed for collecting image statistics characterizing natural images.

2.1.1 QMF Pyramid Decomposition

The first multi-scale image decomposition employed in this work is the QMF pyramid decomposition, based on separable quadrature mirror filters (QMF) [76, 79, 68] (see Appendix A for more detail). One important reason for choosing the QMF pyramid decomposition, while not the more popular wavelet decomposition, is that it minimizes aliasing from the reconstructed image, making it suitable for the purpose of image analysis [1]. Shown in Figure 2.3 (a) is the idealized frequency domain decomposition with a three-scale QMF pyramid (it is idealized as using finite support filters it is not possible to achieve the sharp cut-off in frequency domain as shown). The QMF pyramid decomposition splits the image frequency space into three different scales, and within each scale, into three orientation subbands (vertical, horizontal and diagonal). Visually, each subband captures the local orientation energy in an image. The resulting vertical, horizontal and diagonal subbands at scale i are denoted by $V_i(x, y)$, $H_i(x, y)$, and $D_i(x, y)$, respectively. The first scale subbands are the result of convolving the image with a pair of 1-D $2N + 1$-tap finite impulse response (FIR) low-pass and high-pass QMF filters, denoted as $l(\cdot)$ and $h(\cdot)$ respectively. The vertical subband is generated by convolving the image, $I(x, y)$, with the low-pass filter in the vertical direction and

[1] However, unlike wavelets, the QMF pyramid does not afford perfect reconstruction of the original signal, though the reconstruction error can be made small in practice by careful design of the filters (Appendix A)

the high-pass filter in the horizontal direction as:

$$V_1(x,y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} l(n) I(x-m, y-n). \tag{2.1}$$

The horizontal subband is generated by convolving the image with the low-pass filter in the horizontal direction and the high-pass filter in the vertical direction as:

$$H_1(x,y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} h(n) I(x-m, y-n). \tag{2.2}$$

The diagonal subband is obtained by convolving the image with the high-pass filter in both directions as:

$$D_1(x,y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} h(n) I(x-m, y-n). \tag{2.3}$$

Finally, convolving the image with the low-pass filter in both directions generates the residue low-pass subband, as:

$$L_1(x,y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} l(n) I(x-m, y-n). \tag{2.4}$$

The next scale is obtained by first down-sampling the residual low-pass subband L_1 and recursively filtering with $l(\cdot)$ and $h(\cdot)$, as

$$V_2(x,y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} l(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n) \tag{2.5}$$

$$H_2(x,y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} h(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n) \tag{2.6}$$

$$D_2(x,y) = \sum_{m=-N}^{N} h(m) \sum_{n=-N}^{N} h(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n) \tag{2.7}$$

$$L_2(x,y) = \sum_{m=-N}^{N} l(m) \sum_{n=-N}^{N} l(n) L_1(\lfloor x/2 \rfloor - m, \lfloor y/2 \rfloor - n). \tag{2.8}$$

Subsequent scales are generated similarly by recursively decomposing the residual low-pass subband. The decomposition of a RGB color image is performed by decomposing each color channel independently. These subbands are denoted as $V_i^c(x,y)$, $H_i^c(x,y)$, and $D_i^c(x,y)$, with $c \in \{r, g, b\}$. Color images using other color systems (e.g., HSV or CMYK) are decomposed by first transforming to RGB colors. From this decomposition a series of first- and higher-order statistics are collected (see section 2.2).

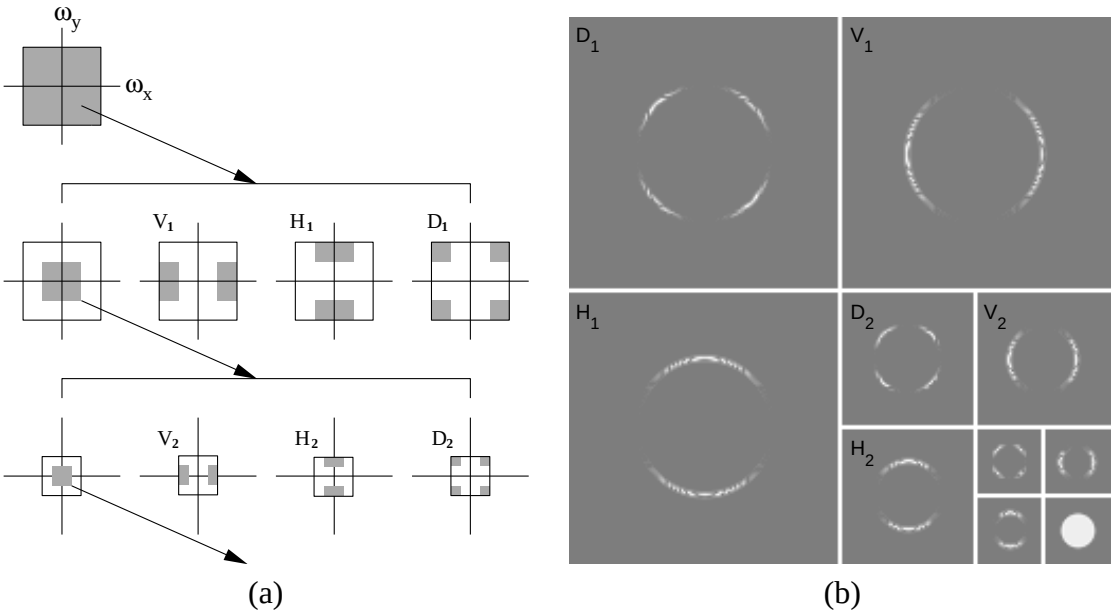


Figure 2.3: (a) An idealized frequency domain decomposition with a three-scale QMF pyramid decomposition. Shown, from top to bottom, are scales 0, 1, and 2, and from left to right, are the low-pass, vertical, horizontal, and diagonal subbands. (b) The magnitude of a three-scale QMF pyramid decomposition of a "disc" image. For the purpose of display, each subband is normalized into range $[0, 255]$.

2.1.2 Local Angular Harmonic Decomposition

The second image decomposition employed in this work is the local angular harmonic decomposition (LAHD) [66]. Formally, the n^{th}-order local angular harmonic decomposition of an image, $I(x, y)$, is defined as:

$$\mathcal{A}_n(I)(x, y) \;=\; \int_r \int_\theta I_{(x,y)}(r, \theta) R(r) e^{in\theta} dr d\theta, \tag{2.9}$$

where $I_{(x,y)}(r, \theta)$ is the polar parameterization of image $I(x, y)$ about point (x, y) in the image plane, and $R(r)$ is an integrable radial function. The LAHD can be regarded as a local decomposition of image structure by projecting onto a set of angular Fourier basis kernels, $e^{in\theta}$. The function $R(r)$ serves as the local windowing function as in the Gabor filters [70], which localizes the analysis in both the spatial and frequency domains. The output of the n-th LAHD, $\mathcal{A}_n(I)(x, y)$, is a complex-valued 2-D signal. Shown in Figure 2.4 are the magnitudes and phases of the first 4-order LAHD of an image. Both the magnitudes and the phases capture image structures such as edges, corners and boundaries. Note that the basis in LAHD is highly over-complete and it is usually not possible to reconstruct the image from the decomposition.

There is a close relation between the LAHD and image derivatives, which is summarized in the following theorem:

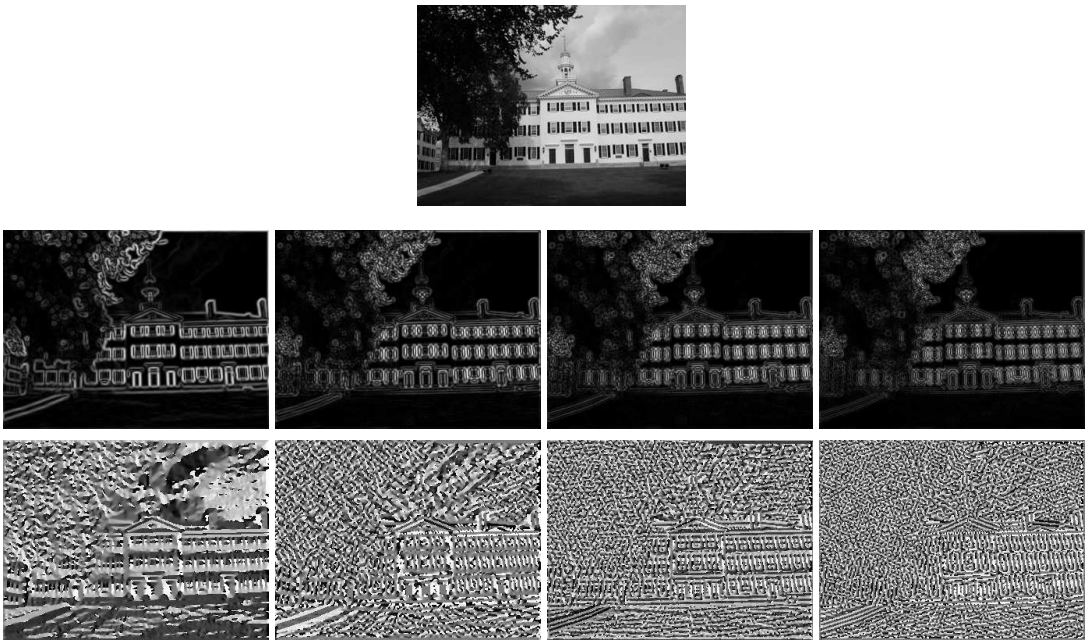


Figure 2.4: The first 4-order LAHD of a natural image (top row). The middle row shows the magnitudes and the bottom row shows the phase angles.

Theorem 1 *Define a 2-D complex-valued signal $\tilde{I}$ as:*

$$\tilde{I}(x,y) \;=\; \left[\left(\frac{\partial}{\partial x}+\imath\frac{\partial}{\partial y}\right)^n (I\star g)\right](x,y), \tag{2.10}$$

where $\star$ denotes the convolution operator. The function $g(\cdot,\cdot)$ is a radial symmetric 2-D filter with up to n^{th}-order derivatives. The power in the definition is denoted as repeated superposition of operators. Then for $R(r)=r\frac{\partial^n g}{\partial r^n}$, we have $\tilde{I}(x,y)=\mathcal{A}_n(I)(x,y)$.

Proof: With the linearity of the convolution to the differential operator, it holds that:

$$\tilde{I}(x,y)=\left[\left(\frac{\partial}{\partial x}+\imath\frac{\partial}{\partial y}\right)^n (I\star g)\right](x,y)=\left[I\star\left(\frac{\partial}{\partial x}+\imath\frac{\partial}{\partial y}\right)^n g\right](x,y).$$

Expanding the 2-D convolution yields:

$$\tilde{I}(x,y) \;=\; \int_\tau\int_\xi I(x-\tau,y-\xi)\left(\frac{\partial}{\partial\tau}+\imath\frac{\partial}{\partial\xi}\right)^n g(\tau,\xi)d\tau d\xi \tag{2.11}$$

Next the image plane is transformed from the 2-D Cartesian coordinates to polar coordinates centered at (x,y), with

$$\tau \;=\; -r\cos\theta \tag{2.12}$$

$$\xi \;=\; -r\sin\theta, \tag{2.13}$$

and $d\tau d\xi = r dr d\theta$. Accordingly, the image is expressed in the polar coordinates as $I_{(x,y)}(r,\theta)$. Solving for $\frac{\partial}{\partial \tau}$ and $\frac{\partial}{\partial \xi}$ in terms of $\frac{\partial}{\partial r}$ and $\frac{\partial}{\partial \theta}$ yields:

$$\frac{\partial}{\partial \tau} = \cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta} \quad (2.14)$$

$$\frac{\partial}{\partial \xi} = \sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta} \quad (2.15)$$

Substituting Equation (2.14) and (2.15) into $\left(\frac{\partial}{\partial \tau} + \imath \frac{\partial}{\partial \xi}\right)^n g(\tau,\xi)$ yields:

$$\begin{aligned}\left(\frac{\partial}{\partial \tau} + \imath \frac{\partial}{\partial \xi}\right)^n g(\tau,\xi) &= \left(\cos\theta \frac{\partial}{\partial r} - \frac{1}{r}\sin\theta \frac{\partial}{\partial \theta} + \imath\left(\sin\theta \frac{\partial}{\partial r} + \frac{1}{r}\cos\theta \frac{\partial}{\partial \theta}\right)\right)^n g(r) \\ &= \left((\cos\theta + \imath\sin\theta)\frac{\partial}{\partial r} - \frac{1}{r}(\sin\theta - \imath\cos\theta)\frac{\partial}{\partial \theta}\right)^n g(r)\end{aligned}$$

Notice that the function $g(r)$ is independent of the parameter θ (it is radially symmetric) - any differentiation of $g(r)$ with respect to θ is zero. Therefore the previous equation can be simplified to:

$$\left(\frac{\partial}{\partial \tau} + \imath \frac{\partial}{\partial \xi}\right)^n g(\tau,\xi) = (\cos\theta + \imath\sin\theta)^n \frac{\partial^n g(r)}{\partial r^n} = e^{\imath n\theta}\frac{\partial^n g(r)}{\partial r^n}, \quad (2.16)$$

with the last step being a result of applying the Euler identity, $e^{\imath\theta} = \cos\theta + \imath\sin\theta$. Substituting Equation (2.16) back into Equation (2.11) yields:

$$\begin{aligned}\tilde{I}(x,y) &= \int_\tau \int_\xi I(x-\tau, y-\xi)\left(\frac{\partial}{\partial \tau} + \imath\frac{\partial}{\partial \xi}\right)^n g(\tau,\xi) d\tau d\xi \\ &= \int_r \int_\theta I_{(x,y)}(r,\theta) r \frac{\partial^n g(r)}{\partial r^n} e^{\imath n\theta} dr d\theta,\end{aligned}$$

and defining $R(r) = r\frac{\partial^n g(r)}{\partial r^n}$ yields:

$$\tilde{I}(x,y) = \int_r \int_\theta I_{(x,y)}(r,\theta) R(r) e^{\imath n\theta} dr d\theta = \mathcal{A}_n(I)(x,y).$$

■

Theorem 1 implies that the LAHD can be computed by convolving the image with a set of derivatives of a smooth radial symmetric filter. For instance, one can use a 2-D Gaussian filter, $g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$. Thus the first-order LAHD, with its real and imaginary parts treated as the x- and y-coordinates, is exactly the gradient of the Gaussian filtered image.

Another important property of the LAHD is on its application to a rotated image. First, denote $R_\alpha\{\cdot\}$ to be the image rotation operator with an angle α. Note that there is no mention of the rotation center. The reason is that in the 2-D image plane, rotating the image region around any two

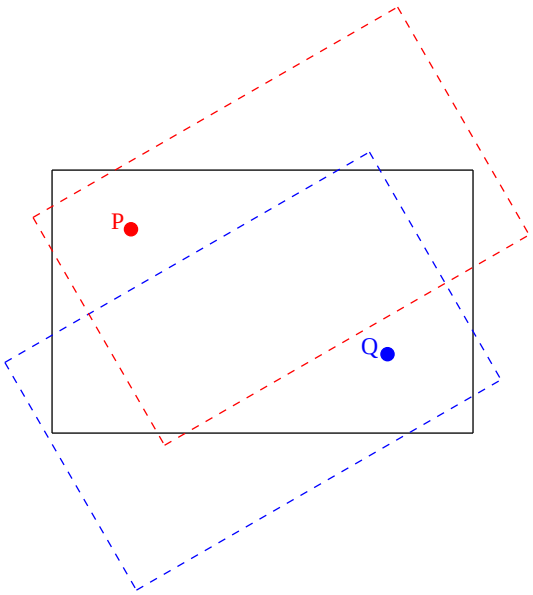


Figure 2.5: Rotating an image around two different points, P and Q, shown as red and blue dashed rectangles, differs by a translation.

points by an angle α differs only by a translation, as illustrated in Figure 2.5. A global translation of image region has no effect on the image signal, and can be easily canceled out by choosing a new origin for the image plane. Therefore, the rotating of an image by an angle α around any point in the 2-D image plane can all be treated as equivalent. For the LAHD of a rotated image, we have the following results:

Theorem 2 *The n^{th}-order LAHD of an image rotated with an angle of α is equivalent to a phase shift of $n\alpha$ for each angular Fourier basis in the LAHD of the original image, as:*

$$\mathcal{A}_n(R_\alpha\{I\})(x,y) \;=\; e^{\imath n\alpha}\mathcal{A}_n(I)(x,y). \tag{2.17}$$

Proof: As the rotation center is irrelevant, we can choose it to be any point in the image plane. Let it be (x,y). The rotated image signal, in terms of polar coordinates centered at (x,y), is $I_{(x,y)}(r,\theta-\alpha)$. Therefore, the (x,y) element of the LAHD of the rotated image is:

$$\begin{aligned}
\mathcal{A}_n(R_\alpha I)(x,y) &= \int_r\int_\theta I_{(x,y)}(r,\theta-\alpha)R(r)e^{\imath n\theta}drd\theta \\
&\overset{\theta'=\theta-\alpha}{=} \int_r\int_{\theta'} I_{(x,y)}(r,\theta')R(r)e^{\imath n(\theta'+\alpha)}drd\theta' \\
&= e^{\imath n\alpha}\int_r\int_{\theta'} I_{(x,y)}(r,\theta')R(r)e^{\imath n\theta'}drd\theta' = e^{\imath n\alpha}\mathcal{A}_n(I)(x,y).
\end{aligned}$$

$\mathcal{A}_1(I)(x,y)$	$I \star \left(\frac{\partial g}{\partial x} + \imath\frac{\partial g}{\partial y}\right)(x,y)$
$\mathcal{A}_2(I)(x,y)$	$I \star \left(\frac{\partial^2 g}{\partial x^2} - \frac{\partial^2 g}{\partial y^2} + 2\imath\frac{\partial^2 g}{\partial x \partial y}\right)(x,y)$
$\mathcal{A}_3(I)(x,y)$	$I \star \left(\frac{\partial^3 g}{\partial x^3} - 3\frac{\partial^3 g}{\partial x \partial y^2} + \imath\left(3\frac{\partial^3 g}{\partial x^2 \partial y} - \frac{\partial^3 g}{\partial y^3}\right)\right)(x,y)$
$\mathcal{A}_4(I)(x,y)$	$I \star \left(\frac{\partial^4 g}{\partial x^4} + \frac{\partial^4 g}{\partial y^4} - 6\frac{\partial^4 g}{\partial x^2 \partial y^2} + \imath\left(4\frac{\partial^4 g}{\partial x^3 \partial y} - 4\frac{\partial^4 g}{\partial x \partial y^3}\right)\right)(x,y)$

Table 2.1: The 1^{st} through 4^{th}-order LAHD computed from image derivatives.

■

In practice, however, computing the LAHD of an image directly by using Equation (2.9) is not recommended. A major concern is that reparameterizing an image from Cartesian coordinates to polar coordinates involves resampling and interpolation of the image plane, which introduce numerical errors. Also, converting to the polar coordinates for each location in an image is inefficient when the image is large in size. On the other hand, the relation between the LAHD and the spatial derivatives revealed by Theorem 1 suggests a more efficient algorithm: the LAHD can be computed by convolving with a set of derivatives of a symmetric and smooth radial filter (e.g., Gaussian). Specifically,

$$\left(\frac{\partial}{\partial x} + \imath\frac{\partial}{\partial y}\right)^n g \;=\; \sum_{k=1}^{n} \binom{n}{k} \imath^{n-k} \frac{\partial^n g}{\partial x^k \partial y^{n-k}}$$

Therefore, according to Theorem 2, the n-th LAHD can be computed as

$$\mathcal{A}_n(I)(x,y) \;=\; I(x,y) \star \sum_{k=1}^{n} \binom{n}{k} \imath^{n-k} \frac{\partial^n g(x,y)}{\partial x^k \partial y^{n-k}} \tag{2.18}$$

Shown in Table 2.1 are the first 4-order LAHD computed as image derivatives.

What has been described is the LAHD for one scale of an image. To capture image information in multiple scales, the LAHD can be computed on each scale of a Gaussian pyramid decomposition of the image, (see section 2.2).

2.2 Image Statistics

The image statistics central to this work are collected from the two image decompositions introduced in the previous section: as statistics of the local magnitudes from the QMF pyramid decomposition and statistics of the local phases from the LAHD. The former capture the characteristics and correlations of local image energy across different scales, orientations and color channels, and

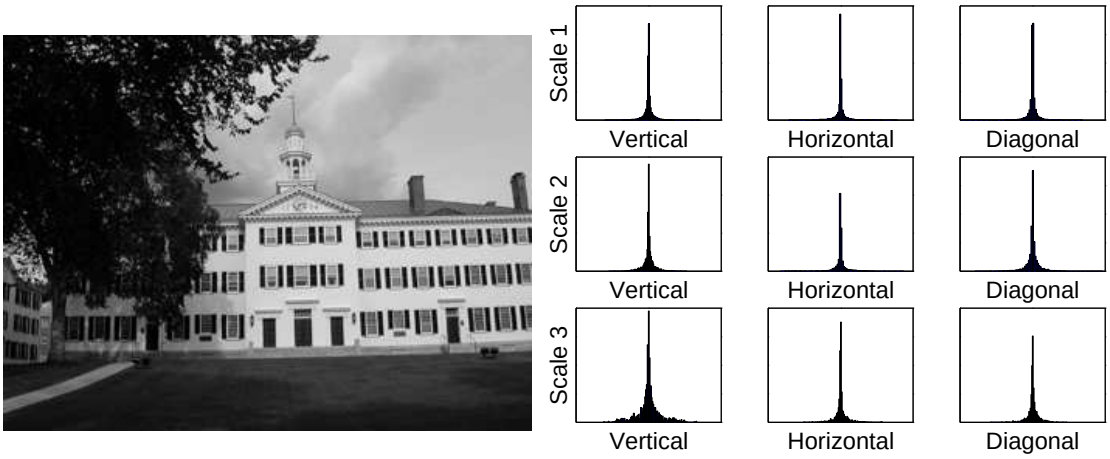


Figure 2.6: A natural image (left) and the histograms of the coefficients of its three-scale QMF decomposition (right). Note the specific shapes of these histograms, characterized by a sharp peak at zero and long symmetric tails.

the latter reveal consistency (or inconsistency) in the local relative phases, which represent the local geometric structure in an image.

2.2.1 Local Magnitude Statistics

We start with the simpler case of grayscale images. It is known that subband coefficients of a multi-scale image decomposition of a natural image have a specific distribution [6], which are characterized by a sharp peak at zero and long symmetric tails, Figure 2.6. An intuitive explanation is that natural images contain large smooth regions and abrupt transitions (e.g., edges). The smooth regions, though dominant, produce small coefficients near zero, while the transitions generate large coefficients. This property holds for the QMF pyramid decomposition as well as many other multi-scale image decompositions (e.g., wavelets).

The marginal distributions of the QMF pyramid subband coefficients exhibit a highly non-Gaussian shape, and usually have positive kurtosis. Many parametric models have been proposed to approximate these distributions. One is the generalized Laplacian distribution [46], $f(x) = \frac{1}{Z}e^{-|x/s|^p}$, where s, p are the density parameters, and Z is a normalizing constant. Another common parametric model is a Gaussian scale mixture [80], where the coefficient is modeled as the product of a zero-mean normal random variable, u and a positive random variable z, as $x = uz$. Then the density of x takes the form as $f(x) = \frac{1}{Z}\int_z \exp(-\frac{x^2}{z^2\sigma_u^2})p_z(z)dz$, where $p_z(\cdot)$ is the density of z, and σ_u^2 is the variance of u. Both models capture the leptokurtic shapes of these distributions, with the model parameters being estimated from training data in a maximum-likelihood fashion.

In this work, instead of modeling these distributions parametrically, a simpler approach is taken. Specifically, we use the first four cumulants (i.e., the mean, variance, skewness and kurtosis, see Appendix B) of the coefficients in each subband of all orientations, scales and color

Figure 2.7: A grayscale natural image (left) and an image reconstructed from the QMF pyramid decomposition of the left image randomly shuffling the decomposition coefficients while keeping the residual low-pass subband. The visual structures in the original image are destroyed though with similar coefficient statistics.

channels to characterize the marginal distributions of the coefficients. The cumulants determine the distribution indirectly - distributions sharing similar cumulants will have similar shapes.

While these statistics describe the basic coefficient distributions, they are not sufficient to characterize a natural image. Specifically, two images that have similar coefficient statistics may be quite different visually. Shown in Figure 2.7 is a simple example, where the left image shares the same residual low-pass subband in a five-scale QMF pyramid decomposition as the right image. However, the coefficients in each subband of the right image are subject to a random shuffling, resulting in the same first order statistics but destroying any higher order correlations in the subbands. Though some overall characteristics of the left image is preserved through the residual low-pass subband (e.g., a darker region on the left and a lighter region on the right), the detailed structures (e.g., edges) in the original image are completely destroyed in the right image.

What causes the problem is the implicit assumption that each coefficient is an independent sample from an underlying distribution. For natural images, this independence assumption is very problematic, as the coefficients in each subband and across different scales and orientations are correlated as the result of salient image features (e.g., edges and junctions) [26]. Such image features tend to orient spatially in certain directions and extend across multiple scales, which result in substantial local energy, measured by the magnitude of the decomposition coefficient, across many scales, orientations and spatial locations, Figure 2.8. As such, a coefficient with a large magnitude in a horizontal subband (an indication that the corresponding pixel is on or near an edge having horizontal orientation) suggests that the left and right spatial neighbors in the same subband may also have a large magnitude. Similarly, if there is a coefficient with a large magnitude at scale i, it is likely that its "parent" at scale $i + 1$ will also have a large magnitude.

This higher-order statistical correlation of coefficients can be captured by another set of image statistics, which are collected from the linear prediction errors of coefficient magnitudes. It has

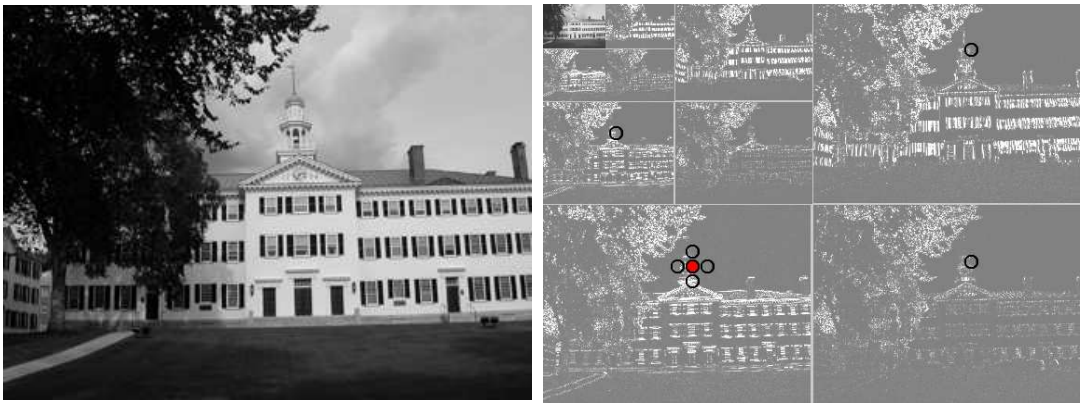


Figure 2.8: A natural image (left) and the magnitudes of its three-scale QMF pyramid decomposition (right). Note the correlation between magnitudes of neighboring coefficients, as highlighted in the right panel and corresponding to the fixed neighborhood set used in Equation 2.19.

been noted that the coefficient magnitudes near image features follow a simple first-order linear dependency [6]. Specifically, it is possible to build a linear predictor of magnitudes from the magnitudes of neighboring coefficients for natural images. The prediction errors from this linear predictor provide a measure of correlations among neighboring coefficients. For the purpose of illustration, consider a coefficient in the vertical subband at scale i, $V_i(x, y)$. A linear predictor of its magnitude is built from a fixed subset, motivated by the observations of [6] and modified to include non-causal neighbors, from its all possible spatial, orientation, and scale neighbors. Formally, the linear predictor of the coefficient magnitudes is formed as a linear combination of the neighboring magnitudes:

$$\begin{aligned} |V_i(x,y)| &= w_1|V_i(x-1,y)| + w_2|V_i(x+1,y)| + w_3|V_i(x,y-1)| + w_4|V_i(x,y+1)| \\ &+ w_5|V_{i+1}(x/2,y/2)| + w_6|D_i(x,y)| + w_7|D_{i+1}(x/2,y/2)|, \end{aligned} \tag{2.19}$$

where $|\cdot|$ denotes the magnitude operator and w_k, $k = 1, ..., 7$ are scalar weights associated with each type of neighbors. Interpolated values (e.g., rounding) are used when $x/2$ or $y/2$ is non-integer. When evaluated throughout the whole subband (and assuming homogeneity), Equation (2.19) can be expressed more compactly in form of matrix and vectors as:

$$\vec{v} = Q\vec{w}, \tag{2.20}$$

where the column vector $\vec{v}$ is formed by stacking the magnitudes of all coefficients in V_i[2], and each column of the matrix Q contains the magnitudes of each type of neighboring coefficients, as specified in Equation (2.19). The unknowns, $\vec{w} = (w_1 \ \ldots \ w_7)^T$, are determined by a least squares estimation, which minimizes the following quadratic error function:

$$E(\vec{w}) = [\vec{v} - Q\vec{w}]^2. \tag{2.21}$$

[2]To make the estimation stable, we only consider coeffi cients with magnitudes greater than a pre-given threshold.

This error function is minimized by differentiating it with respect to $\vec{w}$:

$$\frac{dE(\vec{w})}{d\vec{w}} = 2Q^T(\vec{v} - Q\vec{w}), \tag{2.22}$$

and setting the result equal to zero. Solving the resulting equation for $\vec{w}$ yields:

$$\vec{w} = (Q^TQ)^{-1}Q^T\vec{v}. \tag{2.23}$$

Given the large number of constraints (one per coefficient in the subband) in only seven unknowns, it is generally safe to assume that the 7×7 matrix Q^TQ is invertible. Similar linear predictors are formed on all other orientation subbands - the linear predictor for horizontal and diagonal subbands are given as:

$$\begin{aligned}|H_i(x,y)| = {} & w_1|H_i(x-1,y)| + w_2|H_i(x+1,y)| + w_3|H_i(x,y-1)| + w_4|H_i(x,y+1)| \\ & + w_5|H_{i+1}(x/2,y/2)| + w_6|D_i(x,y)| + w_7|D_{i+1}(x/2,y/2)|,\end{aligned} \tag{2.24}$$

and

$$\begin{aligned}|D_i(x,y)| = {} & w_1|D_i(x-1,y)| + w_2|D_i(x+1,y)| + w_3|D_i(x,y-1)| + w_4|D_i(x,y+1)| \\ & + w_5|D_{i+1}(x/2,y/2)| + w_6|H_i(x,y)| + w_7|V_i(x,y)|.\end{aligned} \tag{2.25}$$

An alternative to the fixed neighborhood subset as used in Equation (2.19) is to use neighboring subsets adapted to different subbands. Specifically, a quasi-optimal neighborhood subset can be found by an iterative greedy search on a per subband and per image basis. The subset of a given number of neighbors that minimizes the prediction error, Equation (2.21), is used to construct the linear predictor of that subband. For a subband at scale i, the search for such an optimal neighborhood subset of size k is constrained to the set of all neighbors within an $N \times N$ spatial region of all orientations at scales $i, i+1, \ldots, i+L-1$. For instance, with $N = 3$ and $L = 3$, the search of the optimal neighborhood subset for vertical subband at scale i, V_i, is confined in the following 80 neighbors:

$$\begin{array}{ccc} V_i(x-c_x, y-c_y) & H_i(x-c_x, y-c_y) & D_i(x-c_x, y-c_y) \\ V_{i+1}(\frac{x}{2}-c_x, \frac{y}{2}-c_y) & H_{i+1}(\frac{x}{2}-c_x, \frac{y}{2}-c_y) & D_{i+1}(\frac{x}{2}-c_x, \frac{y}{2}-c_y) \\ V_{i+2}(\frac{x}{4}-c_x, \frac{y}{4}-c_y) & H_{i+2}(\frac{x}{4}-c_x, \frac{y}{4}-c_y) & D_{i+2}(\frac{x}{4}-c_x, \frac{y}{4}-c_y) \end{array}$$

with $c_x, c_y \in \{-1, 0, 1\}$ and excluding $V_i(x,y)$ itself. The divisions are rounded to integers when necessary. From these neighbors, rather than using an exhaustive search of all possible subsets of size k, a greedy search is employed. On each iteration, a remaining neighbor, whose inclusion minimizes the error function, Equation (2.21), is incorporated. The whole process can be efficiently implemented as an order-recursive least squares estimation. This iterative search process is repeated for all orientation and scale subbands. With the chosen neighborhood subset, the linear predictor is constructed similarly with the predictor coefficients $(w_1, ..., w_k)$ determined as for the fixed neighborhood, Equation (2.23). For natural images, the fixed neighborhood set in Equation (2.19) coincides with the neighborhood found by this quasi-optimal search [6]. On the other

Figure 2.9: A natural image (left) and its reconstruction with the QMF pyramid magnitudes predicted by the linear predictor, Equation (2.19),(2.24) and (2.25), and the signs from the QMF pyramid of the original image. Note the improvement over the image reconstructed with marginal statistics only, Figure 2.7.

hand, for applications other than natural images (e.g., the images of paintings and drawings in the application of art authentication), these adapted neighborhood are more effective.

The linear predictor of QMF coefficient magnitudes can better characterize a natural image. Shown in Figure 2.2.1 is a natural image (left) and its reconstruction with the QMF pyramid magnitudes predicted by the linear predictor, Equation (2.19),(2.24) and (2.25), and the signs from the QMF pyramid of the original image. Note the improvement over the image reconstructed with marginal statistics only, Figure 2.7.

With the linear predictor, the log errors between the actual and the predicted coefficient magnitudes are computed as:

$$\vec{p} \;=\; \log(\vec{v}) - \log(|Q\vec{w}|), \tag{2.26}$$

where the $\log(\cdot)$ is computed point-wise on each vector component. The log error, instead of the least square error, Equation (2.21), is used for a larger dynamic range. As shown in [6], this log error quantifies the correlations of the coefficients in a subband with their neighbors, and natural images tend to have a specific distribution for these errors, Figure 2.10. Following the same rationale as the coefficient statistics, the same statistics (i.e., the mean, variance, skewness and kurtosis) are collected to characterize the error distributions of each scale and orientation subbands.

For a QMF pyramid of n scales, the coefficient marginal statistics are collected for scales $i = 1, ..., n-1$, with a total number of $12(n-1)$ (the mean, variance, skewness and kurtosis for the vertical, horizontal and diagonal subbands in each scale). Similarly, the error statistics are collected at scales $i = 1, ..., n-1$ which also yield a total number of $12(n-1)$. Combining both types of statistics results in a grand total of $24(n-1)$ statistics from a grayscale image.

For a RGB color image, the QMF pyramid decomposition is performed independently on each

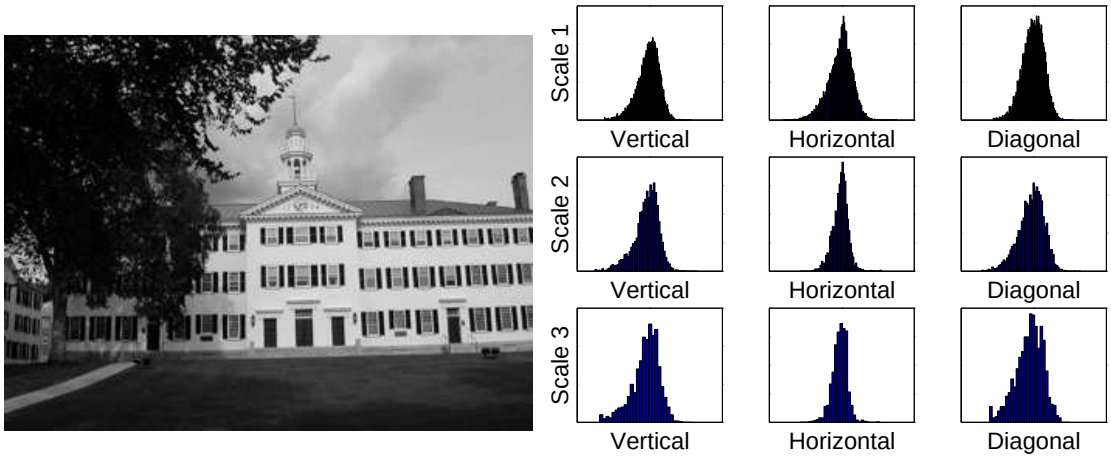


Figure 2.10: A natural image (left) and the histograms of the linear prediction errors of coefficient magnitudes for all subbands in a three-scale QMF pyramid decomposition of the image on the left.

color channel. The coefficient statistics are collected over all subbands in all three color channels and thus triple the total number. Similar to the case of the gray-scale images, a linear predictor is used to capture the correlations among neighboring coefficient magnitudes. Only now the linear predictor is modified to exploit correlations among coefficient magnitudes of different color channels. Specifically, the neighborhood subset used in the linear predictor includes corresponding coefficients in the other color channels[3]. For example, the linear predictor of magnitude for a coefficient in the vertical subband of green channel $V_i^g(x,y)$ is given as:

$$\begin{aligned} |V_i^g(x,y)| &= w_1|V_i^g(x-1,y)| + w_2|V_i^g(x+1,y)| + w_3|V_i^g(x,y-1)| \\ &+ w_4|V_i^g(x,y+1)| + w_5|V_{i+1}^g(x/2,y/2)| + w_6|D_i^g(x,y)| \\ &+ w_7|D_{i+1}^g(x/2,y/2)| + w_8|V_i^r(x,y)| + w_9|V_i^b(x,y)|. \end{aligned} \tag{2.27}$$

This process is repeated for scales $i = 1, ..., n-1$, and for the subbands V_i^r and V_i^b, where the linear predictors for these subbands are of the form:

$$\begin{aligned} |V_i^r(x,y)| &= w_1|V_i^r(x-1,y)| + w_2|V_i^r(x+1,y)| + w_3|V_i^r(x,y-1)| \\ &+ w_4|V_i^r(x,y+1)| + w_5|V_{i+1}^r(x/2,y/2)| + w_6|D_i^r(x,y)| \\ &+ w_7|D_{i+1}^r(x/2,y/2)| + w_8|V_i^g(x,y)| + w_9|V_i^b(x,y)|, \end{aligned} \tag{2.28}$$

and

$$\begin{aligned} |V_i^b(x,y)| &= w_1|V_i^b(x-1,y)| + w_2|V_i^b(x+1,y)| + w_3|V_i^b(x,y-1)| \\ &+ w_4|V_i^b(x,y+1)| + w_5|V_{i+1}^b(x/2,y/2)| + w_6|D_i^b(x,y)| \\ &+ w_7|D_{i+1}^b(x/2,y/2)| + w_8|V_i^r(x,y)| + w_9|V_i^g(x,y)|. \end{aligned} \tag{2.29}$$

[3]As in the case of gray-scale images, the neighbors to form the linear predictor can also be found by a greedy search. For simplicity, only the case of the fixed subset is shown here. Quasi-optimal subsets can be found similarly as for the gray-scale images.

A similar process is repeated for the horizontal and diagonal subbands. As an example, the predictors for the horizontal and diagonal subbands in the green channel are:

$$\begin{aligned} |H_i^g(x,y)| &= w_1|H_i^g(x-1,y)| + w_2|H_i^g(x+1,y)| + w_3|H_i^g(x,y-1)| \\ &+ w_4|H_i^g(x,y+1)| + w_5|H_{i+1}^g(x/2,y/2)| + w_6|D_i^g(x,y)| \\ &+ w_7|D_{i+1}^g(x/2,y/2)| + w_8|H_i^r(x,y)| + w_9|H_i^b(x,y)|, \end{aligned} \tag{2.30}$$

and

$$\begin{aligned} |D_i^g(x,y)| &= w_1|D_i^g(x-1,y)| + w_2|D_i^g(x+1,y)| + w_3|D_i^g(x,y-1)| \\ &+ w_4|D_i^g(x,y+1)| + w_5|D_{i+1}^g(x/2,y/2)| + w_6|H_i^g(x,y)| \\ &+ w_7|V_i^g(x,y)| + w_8|D_i^r(x,y)| + w_9|D_i^b(x,y)|. \end{aligned} \tag{2.31}$$

For the horizontal and diagonal subbands in the red and blue channels, the linear predictors are determined in a similar fashion. For all predictors, the predictor coefficients are estimated similarly with a least squares procedure as in Equations (2.23). For each orientation, scale and color subband, the error metric (Equation (2.26)) is computed, from which the same set of error statistics (mean, variance, skewness and kurtosis) are collected. As three color channels are now being considered, the total number of statistics is tripled to $72(n-1)$ in a QMF pyramid decomposition of n scales.

2.2.2 Local Phase Statistics

Local magnitude statistics from a QMF pyramid decomposition are not sufficient to capture all statistical regularities in natural images. Specifically, one important component that is absent from these statistics is the local phase in image decomposition. It has been known that the phases in a global Fourier transform of an image carry a significant fraction of information in the image. Specifically, structures in an image (e.g., edges, borders and corners) are the results of the precise correlation of phases of different frequency components **??**. Similarly, in a localized image region, the local phases also play a significant role in defining image structures. Capturing such local phase correlations are then important to characterize natural images. It is possible to model local phase statistics from a complex wavelet decomposition [56], affording a unified underlying image representation with the wavelet decomposition described in the previous section. We have found, however, that the local angular harmonic decomposition (LAHD) affords more accurate estimates of local phase [66]. Our image statistics are collected from the rotation-invariant relative phase signatures collected from the LAHD of a natural image.

Relative Phase

From the LAHD of an image, some rotation-invariant signatures can be induced. Consider the relative phases between the m-th and n-th LAHDs of an image I at position (x,y) as [66]:

$$\phi_{mn}(I)(x,y) = \angle\{[\mathcal{A}_n(I)(x,y)]^m, [\mathcal{A}_m(I)(x,y)]^n\}, \tag{2.32}$$

where $\angle\{c_1, c_2\}$ is the angle between two complex numbers c_1 and c_2 in the complex plane, computed as the phase of $c_1 \cdot c_2^*$ ($*$ is the complex conjugate). It is not hard to prove with Theorem 2 that ϕ_{mn} is rotation invariant as:

$$\begin{aligned}\phi_{mn}(R_\alpha I) &= \mathcal{A}_n(R_\alpha I)^m \cdot \mathcal{A}_m(R_\alpha I)^{n*} = e^{\imath mn\alpha}\mathcal{A}_n(I)^m \cdot e^{-\imath mn\alpha}\mathcal{A}_m(I)^{n*} \\ &= \mathcal{A}_n(I)^m \cdot \mathcal{A}_m(I)^{n*} = \phi_{mn}(I).\end{aligned}$$

Furthermore, the magnitudes of the complex LAHDs can also be included to obtain the following rotation invariant [66]:

$$s_{mn} = \sqrt{|\mathcal{A}_n(I)| \cdot |\mathcal{A}_m(I)|} \cdot e^{\imath\phi_{mn}(I)}, \tag{2.33}$$

where all operators (i.e., multiplication, square root, magnitudes and complex exponential) are computed component-wise. From the first N-order LAHDs of an image, $\frac{N(N-1)}{2}$ such rotation invariants, $s_{mn}(I)$ for $1 \leq m < n \leq N$, are collected and used as complex signatures for specific patterns in [66]. For a RGB color image, the LAHDs of each color channel is computed independently, $\mathcal{A}_n^{(c)}(I)$, for $c \in \{r, g, b\}$. The rotation invariants are then computed across color channels as:

$$\phi_{mn}^{(c_1,c_2)}(I)(x,y) = \angle\{[\mathcal{A}_n^{(c_1)}(I)(x,y)]^m, [\mathcal{A}_m^{(c_2)}(I)(x,y)]^n\}, \tag{2.34}$$

and

$$s_{mn}^{(c_1,c_2)} = \sqrt{|\mathcal{A}_n^{(c_1)}(I)| \cdot |\mathcal{A}_m^{(c_2)}(I)|} \cdot \exp(i\phi_{mn}^{(c_1,c_2)}(I)), \tag{2.35}$$

for $c_1, c_2 \in \{r, g, b\}$. In this work, it is from these 2-D complex signals that the second set of statistics of natural images are collected.

Statistics

From the 1^{st}- through N^{th}-order LAHDs, $6N(N-1)$ signatures are collected from within and across color channels (there are $N(N-1)/2$ combinations of the LAHD orders, 6 ordered combinations of color channels, and 2 statistics per combination, yielding $6N(N-1)$). The phase statistics are collected from the two-dimensional distribution of these signatures in the complex plane. Specifically, assuming zero-mean data, we consider the covariance matrix:

$$M_{p,q}^{c_1,c_2} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}, \tag{2.36}$$

where:

$$m_{11} = \frac{1}{S}\sum_{x,y} \Re(s_{p,q}^{c_1,c_2}(x,y))^2 \tag{2.37}$$

$$m_{12} = \frac{1}{S}\sum_{x,y} \Re(s_{p,q}^{c_1,c_2}(x,y))\Im(s_{p,q}^{c_1,c_2}(x,y)) \tag{2.38}$$

$$m_{22} = \frac{1}{S}\sum_{x,y} \Im(s_{p,q}^{c_1,c_2}(x,y))^2 \tag{2.39}$$

$$m_{21} = m_{12}, \tag{2.40}$$

where S is the total number of signatures, and $\Re(\cdot)$ and $\Im(\cdot)$ correspond to the real and imaginary components of a complex quantity. The structure of this covariance matrix is captured by the measures:

$$\mu_1 = \frac{\min(m_{11}, m_{22})}{\max(m_{11}, m_{22})}, \tag{2.41}$$

and,

$$\mu_2 = \frac{m_{12}}{\max(m_{11}, m_{22})}. \tag{2.42}$$

Considering this distribution as a scaled and rotated Gaussian distribution, the first measure corresponds to the relative scales along the minor and major axes, and the second of these measures to the orientation of the distribution. In order to capture these phase statistics at various scales, this entire process is repeated for several scales of a Gaussian pyramid decomposition of the image [50].

2.2.3 Summary

Here we summarize the construction of the statistical feature vector consisted of the local magnitude and local phase statistics from a color (RGB) image.

1. Build a n-scale QMF pyramid decomposition for each color channel, Equations (2.1)-(2.4).

2. For scales $i = 1, \ldots, n-1$ and for orientations V, H and D, across all three color channels, $c \in \{r, g, b\}$, compute the mean, variance, skewness, and kurtosis of the subband coefficients. This yields $36(n-1)$ statistics.

3. For scales $i = 1, \ldots, n-1$, and for orientations V, H and D, across all three color channels, $c \in \{r, g, b]\}$, build a linear predictor of coefficient magnitude, Equation (2.21). From the error in the predictor, Equation (2.26), compute the mean, variance, skewness, and kurtosis. This yields $36(n-1)$ statistics.

4. Build a n-scale Gaussian pyramid for each color channel. For each level of the pyramid, compute the 1^{st}- through N^{th}-order LAHD, Equation (2.18). Compute the relative phases, Equation (2.32). Compute the rotation invariant signature, Equation (2.35), across all color channels and the LAHD orders, from which the covariance matrix, Equation (2.36), and subsequent phase statistics are extracted, Equation (2.41) and (2.42). This yields $6N(N-1)n$ statistics.

5. To avoid the artifact of certain dimensions dominating the classification due to differences in the dynamic range, in the final feature vector, all statistics collected were normalized in each dimension into the range of $[0, 1]$.

In most of the experiments described in this thesis, 216 local magnitude statistics were collected from a three-scale QMF pyramid decomposition of a RGB color image. Similarly, 216 local phase statistics were collected from the 1^{st} to the 4^{th}-order LAHD on a three-scale Gaussian pyramid (using the 5-tap binomial filter $[1\ 4\ 6\ 4\ 1]/16$) from a RGB color image. The overall process of collecting the local magnitude statistics from a QMF pyramid decomposition is summarized in Figure 2.11. It is also possible to collect local magnitude and local phase statistics from grayscale images. With a three-scale QMF pyramid decomposition and the 1^{st} to the 4^{th}-order LAHD on a three-scale Gaussian pyramid, 72 grayscale local magnitude statistics and 36 local phase statistics can be extracted.

2.3 Natural Image Data Set

As our main interest is in statistically modeling natural images, we need a set of natural images in a sufficiently large number and as divergent as possible in their contents. For this purpose, we collected $40,000$ natural images. These images were downloaded from `www.freefoto.com` - all images are from photographs taken with a range of different films, cameras, and lenses, and digitally scanned. They are RGB color images, spanning a range of indoor and outdoor scenes, JPEG compressed with an average quality of 90%, and typically 600×400 pixels in size (on average, 85.7 kilobytes). Grayscale images can be converted from the RGB color images with using mapping: Gray = 0.299Red + 0.587Green + 0.114Blue. Shown in Figure 2.12 are thirty-two examples from this set of natural images. The subsequent experiments described in this thesis are all based on this set of natural images.

2.4 Natural vs. Synthetic

Though based on observations of natural images, the proposed image statistics are not directly derived from first principles of physical imaging process. As such, it is desirable to confirm that they do capture certain non-trivial statistical regularities of natural images. Ideally, under these image statistics, natural images will show more similarity beyond their difference in content, and dis-similarities between natural and un-natural images will also be more distinct. More specifically, in the space of all possible feature vectors consisted of the proposed image statistics, we would like to see that natural images cluster together and separated from un-natural images. It is, however, even harder to define an "un-natural" image, and in our experiments, we employed some synthetic images of different types, usually considered "un-natural" as they rarely appear in natural environment.

Shown in the first row of Figure 2.13 are four examples from $1,000$ natural photographic images chosen from the $40,000$ images as described in Chapter 1. Three different types of synthetic images, Each type of these synthetic images preserving certain statistical properties of natural images, were also generated. Shown in the second to the fourth row in Figure 2.13 are examples of: (1) noise patterns, which were created by scrambling the pixels of the corresponding natural images and thus had the same intensity-histograms; (2) fractal patterns which kept the $1/(\omega_x+\omega_y)^p$,

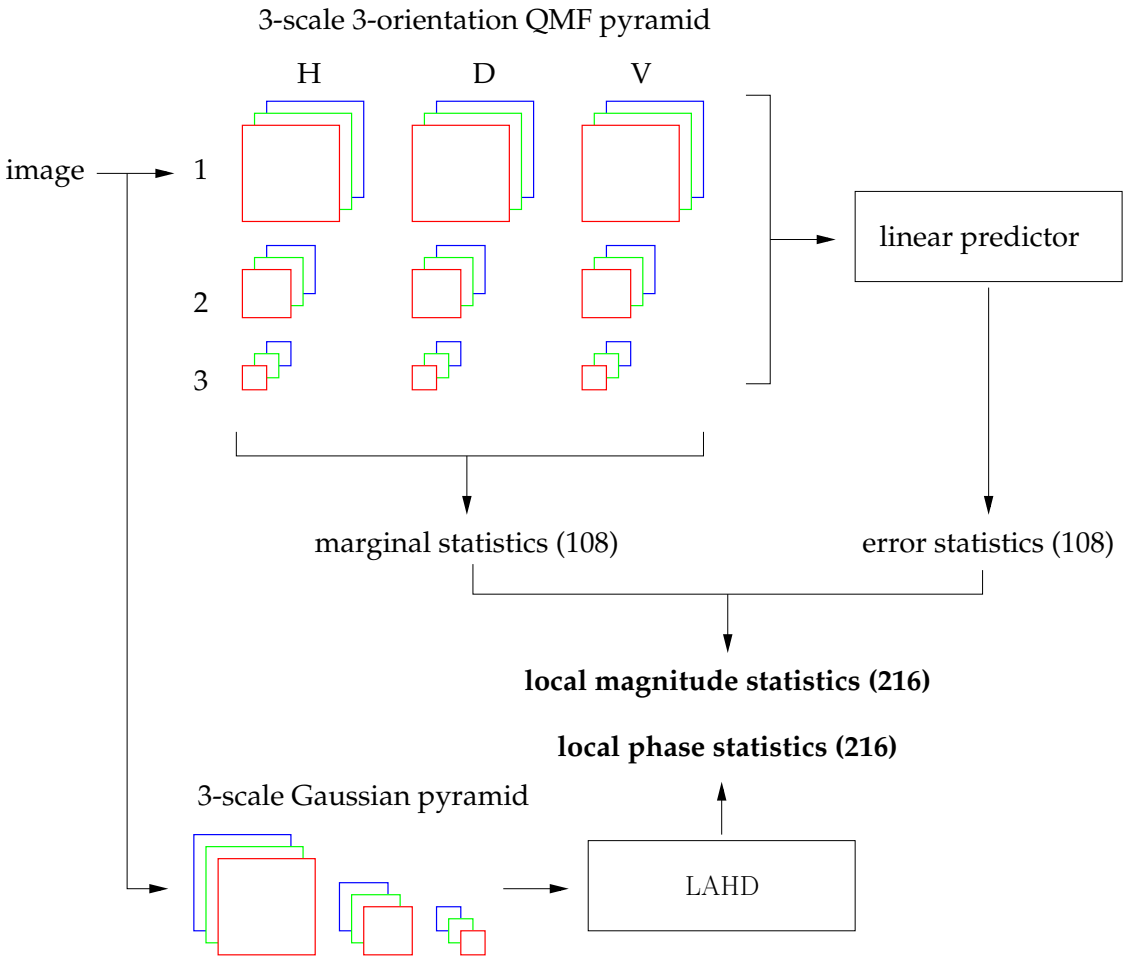


Figure 2.11: Overall process of collecting local magnitude statistics and local phase statistics from a RGB color image with a three-scale QMF pyramid and the 1^{st} to the 4^{th}-order LAHD on a three-scale Gaussian pyramid.

Figure 2.12: Thirty-two examples of natural photographic images from [illegible] of 80,000 natural images.

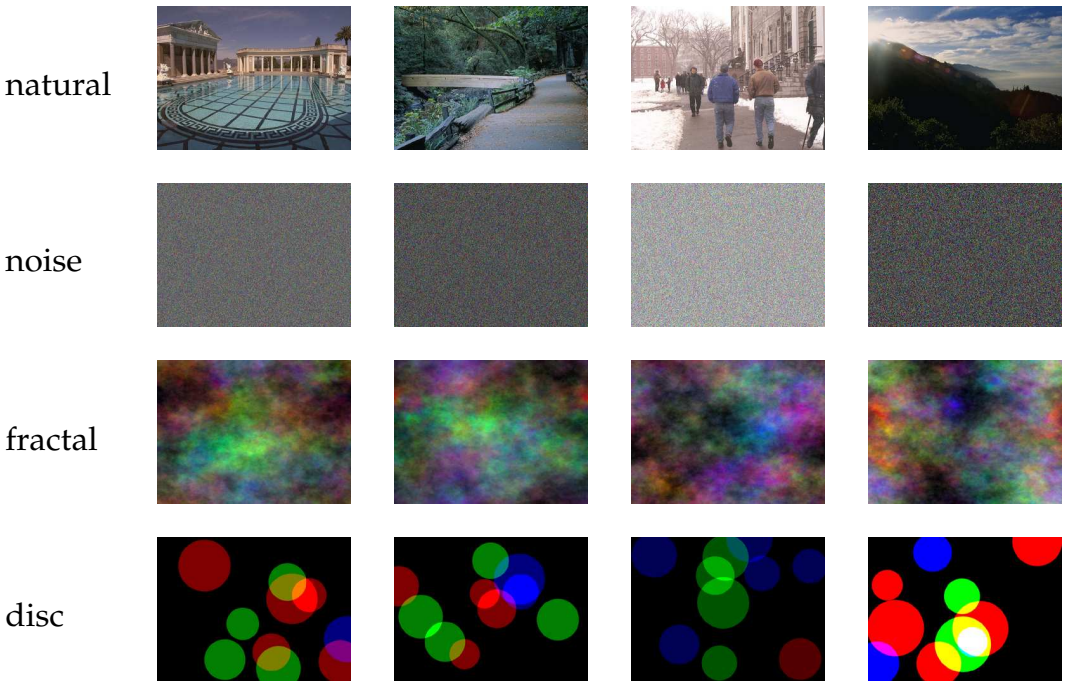


Figure 2.13: Natural and synthetic images. From top to bottom are examples of the $1,000$ natural images, 300 noise patterns with the same intensity-histograms, 300 fractal patterns with the same frequency magnitude envelopes, and 300 disc images with similar phase statistics as the corresponding natural images.

($p \in [1, 2]$) magnitude envelope of the corresponding natural image in the frequency domain but were with random phases; and (3) disc images that were formed by overlapping anti-aliased discs of variable size radii. 300 of each type of synthetic images were generated in our experiment.

From these images, natural and synthetic alike, the local magnitude statistics from the QMF pyramid decomposition and the local phase statistics from the LAHD were extracted to form image features. Specifically, each image was first cropped to the its 256×256 region to accommodate the difference in sizes. Image statistics as described in Section 2.2.3 were collected on each cropped image region. Specifically, three types of image features: (1) the 216 local magnitude statistics, (2) the 216 local phase statistics, and (3) the 432 statistics of both types. To visualize the distribution of these image feature vectors, in all three cases, the high-dimensional feature vectors were projected onto a three dimensional linear subspace spanned by the top three principal components as a result of the principal component analysis of all image features (Appendix C).

Shown in Figure 2.14 are the projected feature vectors of the $1,000$ natural and the 900 synthetic images (300 noise ($\times$), 300 fractal ($\square$), and 300 discs) onto the top 3 principal components for the 216 local magnitude statistics, the 216 local phase statistics, and the 432 statistics of both types. In all three cases, the three top principal components capture over 75% of the total variance in the original data set. By reducing the dimensionality, a significant fraction of information was

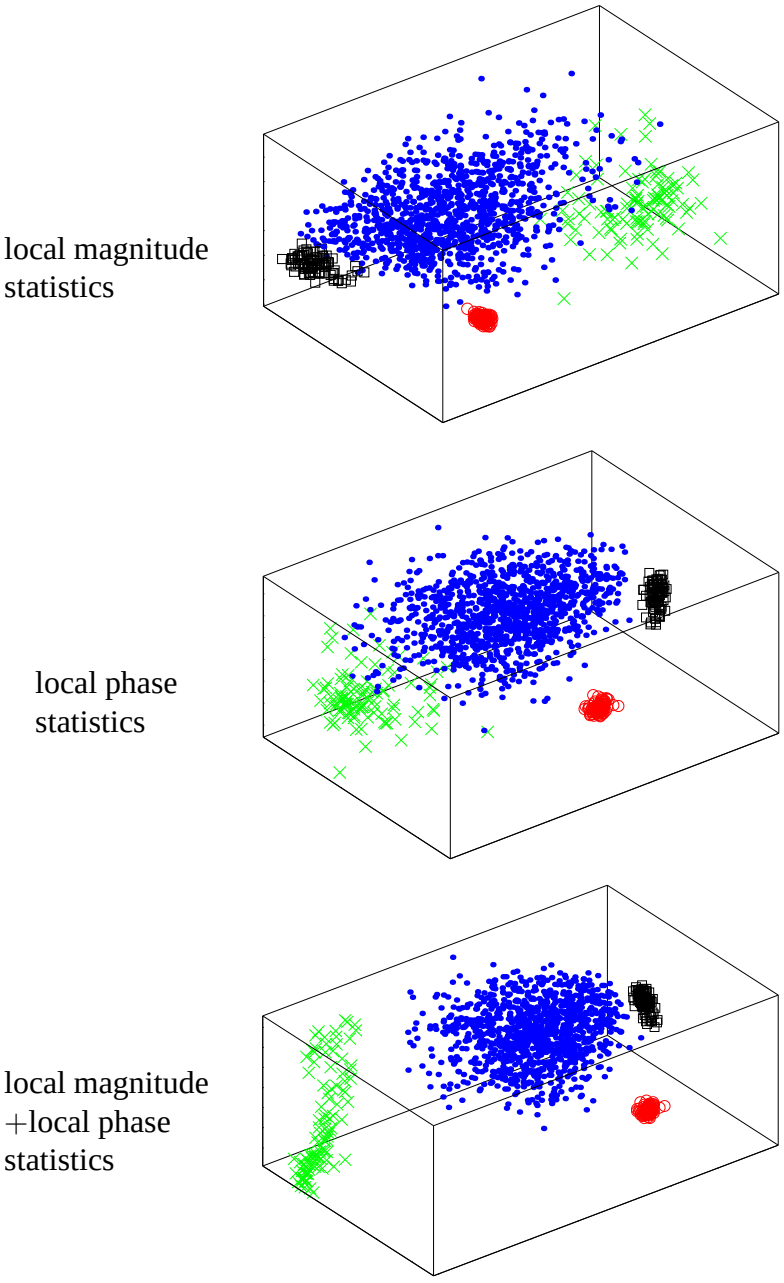


Figure 2.14: Projection of the 216 color local magnitude image statistics, 216 color local phase statistics and 432 color local magnitude and phase statistics for 1000 natural image (•), and the synthetic images (300 noise (×), 300 fractal (□), and 300 discs), onto the top 3 principal components for the 216 local magnitude statistics, the 216 local phase statistics, and the 432 statistics of both types. In all cases, the top three principal components used capture more than 75% of overall variance in the original data set.

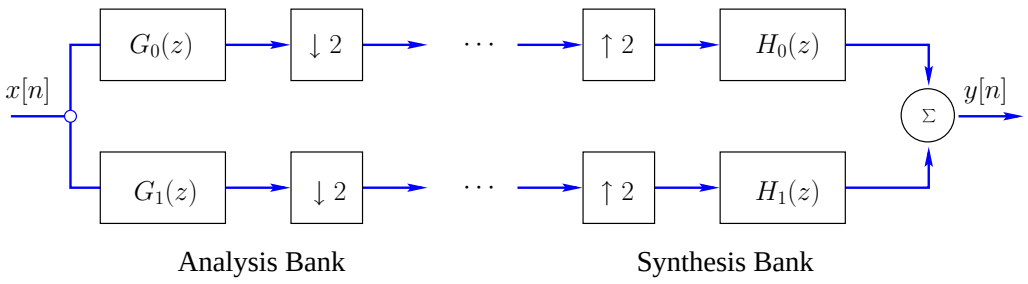


Figure 2.15: Two-channel QMF filter bank.

discarded, but it allows us to visually inspect the distribution of the feature vectors for natural and synthetic images. As we have expected, in all cases, the feature vectors from the proposed image statistics for natural images form a relatively tight cluster. More importantly, the synthesized images are well separated from the ensemble of natural images. The similar statistical regularities in each type of synthetic images are also reflected by their own clusters. These indicate that the proposed image statistics are capable of capturing statistical regularities in natural images not present in the synthetic images. Note that the result with the 432 combined statistics achieved the best separation between natural and synthetic images while keeping a tight ensemble for natural images, suggesting that combining both the local magnitude and the local phase statistics can better characterize natural images.

Appendix A: Quadrature Mirror Filters

In many applications in signal processing, we need to decompose a signal with a two-channel decomposition filter bank, as shown in the following diagram. G_0 and G_1 are low-pass and high-pass filters in the analysis bank, while H_0 and H_1 are low-pass and high-pass filters in the synthesis bank. $\Downarrow 2$ and $\Uparrow 2$ represent the downsampling and upsampling operators, respectively.

$$\begin{aligned} Y(z) &= \left[\frac{G_0(z)H_0(z)+G_1(z)H_1(z)}{2}\right] X(z) \\ &+ \left[\frac{G_0(-z)H_0(z)+G_1(-z)H_1(z)}{2}\right] X(-z). \end{aligned} \tag{2.43}$$

The second term in the above equation is the aliasing in the decomposition. The aliasing cancellation condition (ACC) is expressed as:

$$G_0(-z)H_0(z)+G_1(-z)H_1(z)=0. \tag{2.44}$$

One way to satisfy the ACC is to choose the synthesis filter pairs as:

$$H_0(z)=2G_1(-z) \quad H_1(z)=-2G_0(z). \tag{2.45}$$

With filters satisfying the above equation, the filter bank output in Equation (2.43) is now

$$Y(z) = [G_0(z)G_1(z) - G_0(-z)G_1(z)]\, X(z). \tag{2.46}$$

Another important requirement of the filter bank decomposition is the perfect reconstruction condition (PRC). With Equation (2.43), the PRC is formulated as:

$$G_0(z)H_0(z) + G_1(z)H_1(z) = cz^{-l}, \tag{2.47}$$

for positive constants l and c.

Quadrature mirror filters (QMF) [76, 79, 68] is a design of G_0 and G_1 that satisfy the ACC, i.e., it eliminates aliasing. In QMF, the high-pass filter G_1 is fully determined by the low-pass filter G_0 as:

$$G_1(z) \;=\; G_0(-z). \tag{2.48}$$

It is not hard to see that QMF satisfies the ACC. On the other hand, FIR QMF with more than 2 taps does not satisfy the PRC. For more details, please refer to [70].

Though QMF does not satisfy the PRC, in practice it is still widely used, because it eliminates aliasing and has linear phase response. Furthermore, the amplitude distortion of a QMF in frequency domain is given by:

$$D(\omega) \;=\; 2|G_0(e^{\imath\omega})|^2 - 1. \tag{2.49}$$

Therefore, optimal QMF filter can be designed by finding a linear-phase G_0 minimizes $D(\omega)$ with numerical optimization methods. With such careful design, QMF can give nearly perfect reconstruction. Specifically, in this work, the following pairs of QMF filters are employed:

$$\begin{aligned} l \;=\; & [0.02807382 \;\; -0.060944743 \;\; -0.073386624 \;\; 0.41472545 \\ & \;\; 0.7973934 \;\; 0.41472545 \;\; -0.073386624 \;\; -0.060944743 \;\; 0.02807382], \\ h \;=\; & [0.02807382 \;\; 0.060944743 \;\; -0.073386624 \;\; -0.41472545 \\ & \;\; 0.7973934 \;\; -0.41472545 \;\; -0.073386624 \;\; 0.060944743 \;\; 0.02807382]. \end{aligned}$$

Appendix B: Cumulants

Assume x a real-valued continuous scalar random variable with probability density function $p_x(x)$. The first characteristic function (or moment generating function) of x is defined as the Fourier transform of $p_x(x)$ as:

$$\varphi(\omega) \;=\; \int_{-\infty}^{\infty} p_x(x) e^{\imath\omega x} dx, \tag{2.50}$$

whose McLaughlin expansion yields the moments for x. The second characteristic function (or cumulant generating function) of x is defined by the logarithm of the first characteristic function,

$$\phi(\omega) \;=\; \log(\varphi(\omega)). \tag{2.51}$$

The cumulants are the McLaughlin expansion coefficients of the second characteristic function of x. The first four cumulants of x is defined as:

$$\begin{aligned}
\kappa_1 &= \mathcal{E}\{x\} && (2.52)\\
\kappa_2 &= \mathcal{E}\{x^2\} - (\mathcal{E}\{x\})^2 && (2.53)\\
\kappa_3 &= \mathcal{E}\{x^3\} - 3\mathcal{E}\{x^2\}\mathcal{E}\{x\} + 2(\mathcal{E}\{x\})^3 && (2.54)\\
\kappa_4 &= \mathcal{E}\{x^4\} - 3(\mathcal{E}\{x^2\})^2 + 12\mathcal{E}\{x^3\}\mathcal{E}\{x\} - 6(\mathcal{E}\{x\})^4 && (2.55)
\end{aligned}$$

and more conveniently termed as the mean, variance, skewness and kurtosis of x. Three properties of cumulants make them important in characterizing the random variable x,

1. From cumulants, the pdf of the random variable can be determined.
2. For statistically independent random variables x and y, the cumulants of $x + y$ is the sum of the cumulants of x and y.
3. For multi-dimensional data, the cumulants are multi-linear and captures higher-order correlations.
4. Gaussian distribution has all zero cumulants of order higher than 2.

Appendix C: Principal Component Analysis

Denote column vectors $\vec{x}_i \in \mathcal{R}^n$, $i = 1, ..., N$ as the original feature vectors. The overall mean is:

$$\vec{\mu} = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i \quad (2.56)$$

The zero-meaned data is packed into a $n \times N$ matrix:

$$M = \begin{pmatrix}\vec{x_1} - \vec{\mu} & \vec{x_2} - \vec{\mu} & \dots & \vec{x_N} - \vec{\mu}\end{pmatrix} \quad (2.57)$$

If the dimensionality n of $\vec{x}_i$ is smaller than the number of data points N, as in our case, then the $n \times n$ (scaled) covariance matrix is computed as:

$$C = MM^t \quad (2.58)$$

The principle components are the eigenvectors $\vec{e}_j$ of the covariance matrix (i.e., $C\vec{e}_j = \lambda_j\vec{e}_j$), where the eigenvalue, λ_j is proportional to the variance of the original data along the j^{th} eigenvector. The dimensionality of each $\vec{x}_i$ is reduced from n to p by projecting (via an inner product) each $\vec{x}_i$ onto the top p eigenvalue-eigenvectors. The resulting p-dimensional vector is the reduced-dimension representation.

Chapter 3

Classification

The goal of image classification is to differentiate images of two or more different classes or categories. In digital image forensics applications, the image classes that are of interest are natural photographic images and various types of "un-natural" images, which are subject to operations needed to be revealed for forensics purpose. The image statistics described in the previous chapter are shown to be effective in differentiating natural and such un-natural images. However, such difference in image statistics are very unlikely to be found by visual inspection, as in the case of the experiments in section 2.3. More appropriately, they are formulated as an image classification problem where trained classifiers are employed to differentiate natural images from un-natural images of interest automatically. Automatic classification has been a central theme for pattern recognition and machine learning, and over the years there have been great progress in this direction, see [31] for a general review. For our purpose, we employ three different classification techniques, linear discriminant analysis (LDA), non-linear support vector machines (SVM) and one-class support vector machines (one-class SVM), which will be described in details in this chapter.

3.1 Linear Discriminant Analysis

Strictly speaking, running linear discriminant analysis (LDA) over a set of data does not result in a classifier, but will find a relatively lower dimensional linear data subspace where the classifier can be most easily constructed. More specifically, what LDA achieves is to recover a low-dimensional linear subspace where the classification in the original high-dimensional training data is best preserved. For simplicity a two-class LDA is described, and the extension to multiple classes is straight-forward. Denote d dimensional column vectors $\vec{x}_i$, $i = 1, ..., N_1$ and $\vec{y}_j$, $j = 1, ..., N_2$ as training exemplars from each of two classes. The within-class means are defined as:

$$\vec{\mu}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} \vec{x}_i, \quad \text{and} \quad \vec{\mu}_2 = \frac{1}{N_2} \sum_{j=1}^{N_2} \vec{y}_j. \tag{3.1}$$

The overall-class mean is defined as:

$$\vec{\mu} \;=\; \frac{1}{N_1+N_2}\left(\sum_{i=1}^{N_1}\vec{x}_i+\sum_{j=1}^{N_2}\vec{y}_j\right). \tag{3.2}$$

The within-class scatter matrix is defined as:

$$S_w \;=\; M_1M_1^T+M_2M_2^T, \tag{3.3}$$

where, the i^{th} column of matrix M_1 contains the zero-meaned i^{th} exemplar given by $\vec{x}_i-\vec{\mu}_1$. Similarly, the j^{th} column of matrix M_2 contains $\vec{y}_j-\vec{\mu}_2$. The between-class scatter matrix is defined as:

$$S_b \;=\; N_1(\vec{\mu}_1-\vec{\mu})(\vec{\mu}_1-\vec{\mu})^T+N_2(\vec{\mu}_2-\vec{\mu})(\vec{\mu}_2-\vec{\mu})^T. \tag{3.4}$$

Let $\vec{e}$ be the generalized eigenvector of S_b and S_w with the maximal generalized eigenvalue, that is $S_b\vec{e}=\lambda_{\max}S_w\vec{e}$. The training exemplars $\vec{x}_i$ and $\vec{y}_j$ are projected onto the one-dimensional linear subspace defined by $\vec{e}$ (i.e., $\vec{x}_i^T\vec{e}$ and $\vec{y}_j^T\vec{e}$). This projection simultaneously minimizes the within-class scatter, $\vec{e}^TS_w\vec{e}$, while maximizing the between-class scatter, $\vec{e}^TS_b\vec{e}$, among all such 1-D projections. Once the LDA projection is determined from the training set, a novel exemplar $\vec{z}$ is classified by its projection onto the same subspace, $\vec{z}^T\vec{e}$. In the simplest case, the class to which this exemplar belongs is determined via a simple threshold. In the case of a two-class LDA, we are guaranteed to be able to project onto a one-dimensional subspace, as there will be exactly one non-zero eigenvalue if the within-class means $\vec{\mu}_1$ and $\vec{\mu}_2$ do not collide.

For an N-class classification problem with d-dimensional data, with $N\leq d$, LDA will recover an $(N-1)$-D linear subspace, where the classifiers can be simply constructed as the nearest centroid classifier. In the nearest centroid classification, a data is attributed to the class whose centroid (the geometric center of training class belonging to that class) is closest. In case of the binary classification, where the number of classes $N=2$, the nearest centroid classification equals to a finding a threshold in the 1-D linear subspace (a line) between the two projected class means. The classification surface of the two classes is then a hyperplane orthogonal to the projected direction intersecting it at the threshold point. LDA is attractive because of its general effectiveness and simplicity, as it has a closed-form generalized eigenvector solution. The drawback, however, is that the classification surface is constrained to be linear.

3.2 Support Vector Machines

Support vector machine (SVM) classifiers have recently drawn a lot of attention in pattern recognition and machine learning [7, 78] due both to their solid theoretical foundation and excellent practical performance. In the binary classification, a linear SVM classifier seeks a hyperplane that separates training data of two classes with the largest classification margin, which provably has the best generalization ability (i.e., being able tow working on data not included in the training

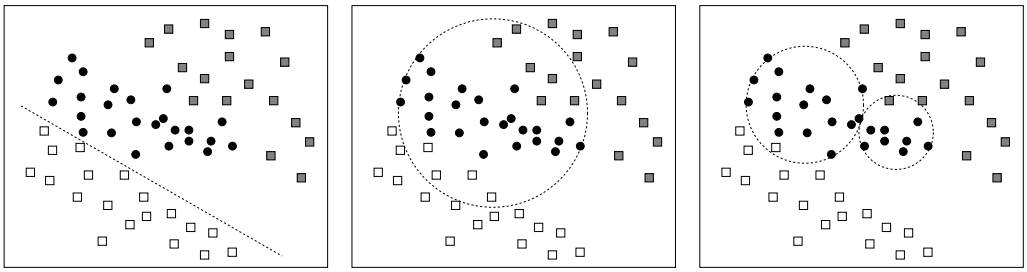
set) among all possible separating hyperplanes. In the case when no hyperplanes possibly separating the training data, SVM finds one that gives rise to the best compromise between classification errors and generalization ability. Using SVM for classification can reduce the risk of overfitting the training data, i.e., the classifier merely memorizing correspondence of training data and class labels, thus will not work on data outside of the training set.

Nevertheless, the real power of SVM lies in its non-linear extension, which reconstructs a non-linear classification surface between the two data classes based on training data. Compared to linear classification techniques such as LDA, being able to use non-linear classification surface greatly increase the flexibility of SVM to model complicated data classification patterns. The non-linear classification, contrary to the arbitrarily complicated non-linear classification techniques such as the neural network, is achieved by first embedding training data into a higher (possibly infinite) dimensional space. A linear separation is then found in that space by the linear SVM algorithm and is mapped back to the original data space as a non-linear classification surface. Such a non-linear classification, though more flexible, inherits the stability and generalization ability of linear SVM, thus effectively reduces the chance of over-fitting the training data. More importantly, the mapping from the original data space to the higher dimensional space, where linear separation found, does not need to be defined explicitly, but can be defined implicitly by computing the inner products of two mapped data via a kernel function (commonly known as the kernel trick). The kernel trick has the advantage that all computations are performed in the original lower-dimensional space. Thus the non-linear SVM does not suffer the potential high dimensionality of the mapped data. The drawback of using non-linear SVM, however, is that its training is more complicated, requiring an iterative numerical optimization and parameter tuning. A more detailed description of the SVM algorithm is given in the Appendix A.

3.3 One-class Support Vector Machines

While non-linear SVMs afford better classification accuracy, its training requires data from all image classes: in our case, the image statistics of both natural and un-natural images. This significantly complicates the data collection and training process. Also, it makes harder to ensure that the classifier to generalize to novel data. Shown in Figure 3.1(a) is a 2-D toy example where a linear SVM classifier, trained on black dots and white squares, is employed to separate dots from squares. The dashed line is the classification surface. In the same figure, the gray squares represent a different type of square data independent of black dots and white squares on which the classifier is trained. The linear SVM classifier cannot correctly classify the gray squares, as the training data provide no information about them. An intuitive explanation is that the classifier pays more attention to the difference between the two classes than the specific properties of one class. One can imagine that in the mapped high dimensional space, non-linear SVM will also suffer from this problem. Possible remedies to this include building another classifier for black dots and gray squares or re-train the existing classifier with all the square data. Either choice involves a larger training set and a more complicated training process.

Another possible solution is to train a classifier using data from one class only. Such a classifier



(a) (b) (c)

Figure 3.1: Shown are toy examples of (a) linear SVM, (b) one-class SVM with one hypersphere, and (c) one-class SVM with two hyperspheres. In each case, the dotted line or circle represents the classifier. The linear SVM is trained on the black dots and white squares. Notice that the gray squares will be incorrectly classified as they were not including in the training. The one-class SVMs are trained on only the black dots. Notice that in these cases the classifier is better able to generalize as both the white and gray squares generally fall outside the support of the bounding circle(s).

classifies by estimating the boundary of the region subsumed by the one class of data used in training. For the image classification problems in digital image forensics, this is possible due to the following fact: to classify natural and un-natural images, it may suffice to know what qualify as natural images, instead of the absolute difference of specific types of un-natural images.

In this aspect, the one-class support vector machines (one-class SVM) [63] is a kernel-based non-linear classification technique whose training set consists of only image statistics of the natural images. A one-class SVM, similar to a non-linear SVM, first projects training data into a higher (potentially infinite) dimensional space implicitly through a proper kernel function. Then a bounding hypersphere in that space is computed such that it encompasses as much of the training data as possible, while minimizing its volume. In the testing stage, only data that fall inside the estimated boundary are considered to be of the same class as training data (see Appendix B for more details). Shown in Figure 3.1(b) is an one-class SVM classifier trained on the black dots in the 2-D toy example. All types of squares are reasonably well separated from the dots by the classifier.

One-class SVM with Multiple Hyperspheres

One potential drawback of using only a single hypersphere in an one-class SVM classifier, however, is that it may not provide a particularly compact estimation of the boundary. As shown in Figure 3.1(b) the bounding hypersphere computed includes also many data from the other classes. To alleviate this problem, we propose to cover the training data with several hyperspheres, where each hypersphere encompasses a non-intersecting subset of the training data. Shown in Figure 3.1(c), for example, is the result of using two hyperspheres to cover the same data as shown in panel (b). Note that, in this case, the subspace of dot data estimated is significantly more compact, leading to improved classification. In choosing the number of hyperspheres, however, we need to balance between the compactness of the subspace and the generalization ability of the classifier. Specifically, if too many hyperspheres are used, then it is likely that the classifier will be tuned only to

the training data, and will perform poorly when presented with novel data.

With a specified number of hyperspheres, M, the training data are first automatically segmented into M non-intersecting subsets. Specifically, a standard K-means clustering algorithm [15] is employed to cluster the original data into M groups. An one-class SVM, using a single hypersphere, is then independently trained on each of the M groups. We next compute the distance between each data point and the center of each one-class SVM's hypersphere. For each data point, the hypersphere whose center is closest is determined. Each data point is then re-assigned to this group, regardless of its previous assignment. And finally, a new set of M one-class SVMs are trained using the new group assignments. This process is repeated until no single data point is re-assigned. The convergence of this algorithm can be proven in a fairly straight-forward way similar to that used in proving the convergence of K-means clustering. In the testing stage, a novel image is tested against each of the M one-class SVMs. It is classified as a natural image if it falls within the support of any one-class SVM's hypersphere, otherwise it is classified as an un-natural image.

Appendix A: Support Vector Machines

A.1: Linear SVM on Linearly Separable Data

Denote the tuple $(\vec{x}_i, y_i)$, $i = 1, ..., N$ as training exemplars from two classes with $\vec{x}_i \in \mathcal{R}^d$ and $y_i \in \{-1, +1\}$ the class labels. The data are linearly separable if a hyperplane exists that separates the two classes. More specifically, there exists a hyperplane

$$\vec{w}^t\vec{x}_i + b \;\; = \;\; 0, \tag{3.5}$$

such that within a scale factor:

$$\vec{w}^t\vec{x}_i + b \;\; \geq \;\; +1, \qquad \text{if } y_i = +1 \tag{3.6}$$

$$\vec{w}^t\vec{x}_i + b \;\; \leq \;\; -1, \qquad \text{if } y_i = -1. \tag{3.7}$$

These linearly separable constraints can be combined into a single set of inequalities:

$$y_i(\vec{w}^t\vec{x}_i + b) - 1 \geq 0, \qquad i = 1, ..., N. \tag{3.8}$$

Among all hyperplanes that can linearly separate the training data, a linear SVM seeks the one with the maximal margin, defined as $2/||\vec{w}||$, with $||\cdot||$ denoting the l_2 norm. Equivalently, this is the solution to the following quadratic optimization problem:

$$\min_{\vec{w},b} \;\; \tfrac{1}{2}||\vec{w}||^2 \tag{3.9}$$

subject to the linear separable constraints, (3.8).

For reasons to be clear later, this optimization problem is reformulated into its dual form. This is achieved by first forming the Lagrangian:

$$L(\vec{w}, b, \alpha_1, ..., \alpha_N) \;\; = \;\; \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{N} \alpha_i \, y_i \, (\vec{w}^t\vec{x}_i + b) + \sum_{i=1}^{N} \alpha_i, \tag{3.10}$$

where α_i are the non-negative Lagrange multipliers. We then differentiate $L(\vec{w}, b, \alpha_1, ..., \alpha_N)$ with respect to $\vec{w}$ and b, and set the results equal to zero to yield:

$$\vec{w} = \sum_{i=1}^{N} \alpha_i \vec{x}_i y_i \tag{3.11}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0. \tag{3.12}$$

Substituting these equalities back into Equation (3.10) yields the dual problem:

$$\max_{\alpha_1,\ldots,\alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i^t \vec{x}_j \tag{3.13}$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.14}$$

and

$$\alpha_i \geq 0, \qquad i = 1, \ldots, N. \tag{3.15}$$

It can be proved that the maximum of this dual problem equals to the minimum of our primary problem, which seeks the separating hyperplane with the maximal margin. This means that the primary problem can be solved equivalently with the dual problem. Any general purpose optimization package that solves linearly constrained convex quadratic problems (see e.g., [17]) can be employed for that purpose, though procedures specifically designed for this task may be more efficient.

A solution to the dual problem, (3.13)-(3.15), yields optimal values of α_i, from which $\vec{w}$ can be calculated as in Equation (3.11). Those data with strictly positive Lagrangian multipliers are called support vectors, from which b is computed as:

$$b \;=\; \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \vec{w}^t \vec{x}_i \right), \tag{3.16}$$

for all i, such that $\alpha_i \neq 0$. From the separating hyperplane, $\vec{w}$ and b, for a novel exemplar, $\vec{z}$, its class label is set to $sgn(\vec{w}^t \vec{z} + b)$, where

$$sgn(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}. \tag{3.17}$$

A.2: Linear SVM on Linearly Non-separable Data

For data that are not linearly separable, the constraints to the maximal margin problem are modified with "slack" variables, ξ_i, as follows:

$$\vec{w}^t\vec{x}_i + b \geq +1 - \xi_i, \qquad \text{if } y_i = +1 \tag{3.18}$$

$$\vec{w}^t\vec{x}_i + b \leq -1 + \xi_i, \qquad \text{if } y_i = -1, \tag{3.19}$$

with $\xi_i \geq 0,\ i = 1, ..., N$. A training exemplar which lies on the "wrong" side of the separating hyperplane will have a value of ξ_i greater than unity. We then seek a hyperplane that minimizes the total training error, $\sum_{i=1}^{N} \xi_i$, while still maximizing the margin. This is formulated as the following objective function:

$$\frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{N} \xi_i, \tag{3.20}$$

where C is a user selected scalar value, whose chosen value controls the relative penalty for training errors. Minimization of this objective function with constraints (3.18) is a quadratic programming problem. Following the same procedure as the previous section, the dual problem is expressed similarly as maximizing the objective function Equation (3.13) with constraints (3.14) and $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, N$. Note that this is the same optimization problem with the slightly different constraint that α_i is bounded above by C. Again it can be solved numerically and the computation of the hyperplane parameters is accomplished as described in the previous section.

A.3 Non-Linear SVM

Fundamental to the linear SVMs outlined in the previous two sections is the limitation that the classifier is constrained to a linear hyperplane. One can imagine that data not linearly separable may be separated by a non-linear surface, yielding a non-linear decision function. Non-linear SVMs afford such a classifier by first mapping the training exemplars into a higher (possibly infinite) dimensional inner product space in which a linear SVM is then employed.

Denote this mapping as:

$$\phi : \mathcal{R}^d \rightarrow \mathcal{F}, \tag{3.21}$$

which maps the original training data from $\mathcal{R}^d$ into $\mathcal{F}$. Replacing $\vec{x_i}$ with $\phi(\vec{x_i})$ everywhere in the training portion of the linear separable or non-separable SVMs of the previous sections yields an SVM in the higher-dimensional space $\mathcal{F}$.

It can, unfortunately, be quite inconvenient to work in the space $\mathcal{F}$ as this space can has a considerably higher or even infinite dimension. Note, however, that the objective function of the dual problem, Equation (3.13), depends only on the inner products of the training exemplars, $\vec{x}_i^t\vec{x}_j$. Given a kernel function such that:

$$k(\vec{x}, \vec{y}) = \phi(\vec{x})^t\phi(\vec{y}), \tag{3.22}$$

an explicit computation of ϕ can be completely avoided. There are several choices for the form of the kernel function, for example, radial basis functions or polynomials. Replacing the inner products $\phi(\vec{x}_i)^t\phi(\vec{x}_j)$ with the kernel function $k(\vec{x}_i, \vec{x}_j)$ yields an SVM in the space $\mathcal{F}$ with minimal computational impact over working in the original space $\mathcal{R}^d$.

With the training stage complete, a novel exemplar, $\vec{z}$, is classified by $sgn(\vec{w}^t\phi(\vec{z}) + b)$. Only that $\vec{w}$ and $\phi(\vec{z})$ are now in the space $\mathcal{F}$. As in the training stage, the classification can again be performed via inner products which will be replaced by kernel function evaluations. From Equation (3.11), we have

$$\begin{aligned} \vec{w}^t\phi(\vec{z}) + b &= \sum_{i=1}^{N} \alpha_i y_i \phi(\vec{x}_i)^t \phi(\vec{z}) + b \\ &= \sum_{i=1}^{N} \alpha_i y_i k(\vec{x}_i, \vec{z}) + b, \end{aligned} \tag{3.23}$$

which corresponds to a non-linear classification surface.

Appendix B: One-Class Support Vector Machines

Consider N training exemplars in a d-dimensional space denoted as $\{\vec{x}_1, ..., \vec{x}_N\}$. An one-class SVM first projects these data into a higher, potentially infinite, dimensional space with the mapping: $\phi : \mathcal{R}^d \rightarrow \mathcal{F}$. In this space, a bounding hypersphere is computed that encompasses as much of the training data as possible, while minimizing its volume. This hypersphere is parameterized by a center, $\vec{c}$, and a radius, r. Described below is how these parameters are computed from the training data, and then how classification is performed given this bounding hypersphere.

The hypersphere center $\vec{c}$ and radius r are computed by minimizing:

$$\min_{\vec{c}, r, \xi_1, ..., \xi_N} r^2 + \frac{1}{N\nu} \sum_{i=1}^{N} \xi_i, \tag{3.24}$$

where $\nu \in (0, 1)$ is a parameterized constant that controls the fraction of training data that fall outside of the hypersphere, and ξ_is are the "slack variables" whose values indicate how far these outliers deviate from the surface of the hypersphere. This minimization is subject to:

$$\|\phi(\vec{x}_i) - \vec{c}\|^2 \leq r^2 + \xi_i, \quad \xi_i \geq 0, \quad i = 1, ..., N, \tag{3.25}$$

where $\| \cdot \|$ is the Euclidean norm. The objective function of Equation (3.24) embodies the requirement that the volume of the hypersphere is minimized, while simultaneously encompassing as much of the training data as possible. The constraints, (3.25), force the training data to either lie within the hypersphere, or closely outside its surface, with the distance being a positive slack variable ξ_i.

To determine $\vec{c}$ and r, the quadratic programming problem of Equations (3.24)-(3.25) are transformed into their dual form:

$$\min_{\alpha_1,...,\alpha_N} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j\phi(\vec{x}_i)^T\phi(\vec{x}_j) - \sum_{i=1}^{N} \alpha_i\phi(\vec{x}_i)^T\phi(\vec{x}_i), \tag{3.26}$$

subject to:

$$\sum_{i=1}^{N} \alpha_i = 1, \qquad 0 \leq \alpha_i \leq \frac{1}{N\nu}, \qquad i = 1, ..., N, \tag{3.27}$$

where α_i's are Lagrange multipliers. Note that in this dual formulation the constraints (3.27) are now linear, and both the objective function and constraints are convex. Standard techniques from quadratic programming can be used to solve for the unknown Lagrange multipliers [17]. The center of the hypersphere, is then given by:

$$\vec{c} \;\; = \;\; \sum_{i=1}^{N} \alpha_i\phi(\vec{x}_i). \tag{3.28}$$

In order to computed the hypersphere's radius, r, we first use the Karush-Khun-Tucker (KKT) condition [17] to find the data points that lie exactly on the surface of the optimal hypersphere. Such points, $\vec{x}_i$, satisfy the condition $0 < \alpha_i < 1/(n\nu)$. Any such data point $\vec{y}$ that lies on the surface of the optimal hypersphere satisfies the following:

$$r^2 \;\; = \;\; \|\phi(\vec{y}) - \vec{c}\|^2. \tag{3.29}$$

Substituting the solution of Equation (3.28) into the above yields a solution for the hypersphere radius:

$$r^2 \;\; = \;\; \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j\phi(\vec{x}_i)^T\phi(\vec{x}_j) - 2\sum_{i=1}^{N} \alpha_i\phi(\vec{x}_i)^T\phi(\vec{y}) + \phi(\vec{y})^T\phi(\vec{y}). \tag{3.30}$$

With the hypersphere parameters, the decision function, $f(\vec{x})$, which determines whether a data point lies within the support of the hypersphere, is defined as:

$$f(\vec{x}) \;\; = \;\; r^2 - \|\phi(\vec{x}) - \vec{c}\|^2, \tag{3.31}$$

such that, if $f(\vec{x})$ is greater than or equal to zero, then $\phi(\vec{x})$ lies within the hypersphere, otherwise it lies outside. Substituting the solution of Equation (3.28) into the above decision function yields:

$$f(\vec{x}) = r^2 - \left(\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j\phi(\vec{x}_i)^T\phi(\vec{x}_j) - 2\sum_{i=1}^{N} \alpha_i\phi(\vec{x}_i)^T\phi(\vec{x}) + \phi(\vec{x})^T\phi(\vec{x}) \right). \tag{3.32}$$

Note that both the training and classifying process require an explicit evaluation of $\phi(\vec{x})$. This is computationally costly if $\phi(\cdot)$ maps the data into a very high dimensional space, and is problematic when that space is infinite dimensional. However, similar to non-linear SVM, the evaluation of $\phi(\cdot)$ can be avoided entirely by introducing a kernel function, (3.22). The inner products between two projected data points in the above equation, in the computation of r of Equation (3.30), and the objective function of Equation (3.26) are replaced with evaluations of the kernel functions to yield:

$$f(\vec{x}) = \left(2\sum_{i=1}^{N}\alpha_i k(\vec{x}_i,\vec{x}) + k(\vec{x},\vec{x})\right) - \left(2\sum_{i=1}^{N}\alpha_i k(\vec{x}_i,\vec{y}) + k(\vec{y},\vec{y})\right), \tag{3.33}$$

where the re-formulated objective function takes the form:

$$\min_{\alpha_1,\ldots,\alpha_N} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j k(\vec{x}_i,\vec{x}_j) - \sum_{i=1}^{N}\alpha_i k(\vec{x}_i,\vec{x}_i). \tag{3.34}$$

Note that this objective function is now defined in the original d-dimensional space, which obviates computation in the high-dimensional space.

Chapter 4

Sensitivity Analysis

Most digital images may have undergone some operations without the viewer's knowledge, either innocuous or malicious. Such operations range from the simple cropping or rotations to sophisticated doctoring of image contents. Images may also come with noise and artifacts from the sensory or transmission process (e.g., the blocky artifact as a result of JPEG compression). These operations and artifacts (hereafter collectively termed as image manipulations) perturb the image signal, and the image statistics as well.

In certain circumstances, however, these image manipulations are not of central interest and thus irrelevant. For instance, in digital image forensics, small amount of additive image noise due to transmission should not be treated in the same way as a hidden message. Yet if no consideration of such irrelevant image manipulation is taken, the classification with image statistics will result in many false positives (i.e., false alarms). More appropriately, it is desirable that such irrelevant image manipulations are detected and then removed from the training set of the classifier, and in classification, rejected by the classifier. It is therefore of our interest to analyze the sensitivity of the proposed image statistics under some common image manipulations, which are important for both improving the classification performance and simplifying the overall training process.

Specifically, we would like to investigate under what image manipulations the proposed image statistics and classification based upon are affected, and if so, to what degree they are affected. To this end, we take an empirical methodology. First a one-class support vector machine (one-class SVM) classifier was built on the proposed images statistics of a large set of natural images. Using the learned one-class SVM classifier as a model of the image statistics of natural images, we then able to study how the image statistics of the manipulated images deviate from those of the natural images in terms of classification accuracy.

4.1 Training

The training set of the one-class SVM classifier was formed by the $40,000$ natural photographic images as described in Chapter 1. Manipulated images were generated by performing six common image manipulations on these natural images, Figure 4.1 and 4.2:

1. **Cropping:** The size of an image can be changed by cropping, which also affect the image statistics, as natural image signals are not spatially stationary. For simplicity, we tested only central cropping, where the central image regions of the same aspect ratio as the original image but of different sizes were kept. The cropping is parameterized by the cropping ratio, which is the ratio between the cropped area and the original image region. Specifically, we generated cropped images with cropping ratio ranging from 0.5 to 0.9 with a step size of 0.1.

2. **Rotation:** Images can also be change with rotation. By nature, rotation is an operation in the continuous image region. Numerically, it is implemented with interpolation. Rotation introduces perturbation into the image statistics. Just consider a special case of a 90° rotation, all the local magnitude statistics of the horizontal and vertical subbands flip locations. We created rotated images with various rotation angles counter-clockwise around the geometrical center, from 0° to 90° with a step size of 5°.

3. **JPEG compression:** Changes in image formats or format-related parameters are another major source of perturbations to the image statistics. For JPEG images, the most important parameter is the JPEG quality. A commonly used JPEG quality metric is the IJTG standard, which are subjective qualities between 0 and 100 measured from psychophysical experiments, with 100 being the highest quality and 0 the worst. Different JPEG qualities stipulate different quantization tables and the truncation lengths used in compression. Low quality JPEG image has two distinct artifacts: the false block edges due to the 8×8 block segmentation in compression, and the loss of high frequency components due to quantization. These artifacts result in perturbations in the image statistics. We created, from the natural images, JPEG images of different JPEG qualities, from 10 to 90 with a step size of 10.

4. **Noise:** All images carry noise, differing only in the degrees. There are various sources of image noises from image sensors, transmission and compression. A common model of image noise is to assume the noise are additive and independent from the image, with independent multivariate Gaussian distribution (such noise is called white Gaussian noise). The amount of image noise is measured by the signal-to-noise ratio (SNR), which is defined as $SNR = 20 \log_{10}(\text{std(signal)}/\text{std(noise)})$, where std is the standard deviation operator. The unit of the SNR is deci-bell. In our experiments, white noise of different SNR (ranging from 1dB to 100dB with 10 uniform steps in the log space) were added to natural images to generate noise corrupted images.

5. **Blurring:** When the high frequency components in an image are modulated, visually, the image will look blurry. The blurring operation can be implemented by convolving the image with a low-pass filter. Blurring is a common manipulation for anti-aliased rendering of an image. We generated blurred image by convolving the original image with a radially symmetric Gaussian filter, defined as $G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-(x^2 + y^2)/2\sigma^2)$. The width of the filter, σ, controls the degree of blurring. The larger its value is, the more blurred is the image. Specifically, we chose σ from 0.3 to 3.0 with a step size of 0.3.

6. **Phase perturbation:** It has been known that phases in the Fourier transform of an image carry most of the structural information - image structures such as edges are the results of phase correlations. We generated phase perturbed images by randomizing phases while keeping the magnitudes. An image was first subject to a Fourier transform. Then the phases in a designated frequency range were randomly shuffled. To ensure that the reconstructed phase-perturbed image is still real-valued, the shuffled phases were kept anti-symmetric. The phase perturbed image was reconstructed by taking the inverse Fourier transform of the modified frequency responses. The degree of phase perturbation was decided by a parameter r, ranging from 0 to 1 with a step size of 0.1. The parameter r makes the lower $2^{-10(1-r)}$ fraction of frequencies unaffected.

From each natural image and each manipulated image, the 432 image statistics consisting of the 216 local magnitude statistics and the 216 local phase statistics were collected (Section 2.2.3). A one-class SVM classifier with six hypersphere's was trained on the $40,000$ natural images and tested on the manipulated images. Instead of using a hard threshold classifier, we computed the distance from the image statistics to the closest hyper-sphere in the one-class SVM. A positive distance indicates the corresponding image statistics reside inside the hyper-spheres and is classified as from the same class of the training data. A negative distance suggests that they are outside the hypersphere and subsequently classified as different from training data. Intuitively, the larger the absolute value of the distance is, the further away the feature vector is from the classification surface and therefore the less ambiguous the classification of the corresponding image.

4.2 Classification

Shown in Figure 4.3 are the one-class SVM classifier tested on the manipulated images, with panels (a)-(f) corresponding to cropping, rotation, JPEG compression, noise, blurring and phase perturbation, respectively. The horizontal axes in each panel correspond to the parameters of each manipulation, i.e., cropping ratios, rotating angles, JPEG qualities, SNRs, blurring filter widths and the parameter r controlling unaffected frequency region in phase perturbation. The vertical axes correspond to the minimum distance to the classification surface found by the one-class SVM classifier with six hyperspheres. Shown in the plots as dotted lines are the classification threshold (a zero distance to the classification surface). For each manipulation and manipulation parameter, we show both the mean distances to the classification surface of the one-class SVM averaged over all manipulated images (solid line) and the corresponding standard deviations (error bars). Also shown in the plots are the corresponding histograms of these distances as shaded blocks in background, where the grayscales are proportional to the probability mass in the corresponding bins, with a white color indicating a zero probability.

Our experiments suggested that the proposed image statistics were affected by the tested image manipulations, and the influence of these image manipulations on the image statistics and the one-class SVM classification increase as the manipulated images were more degraded from the natural image. Specifically, for image cropping, the general trend from the classification results is that, the smaller the cropped region is, the more unnatural the image is according to the one-class SVM

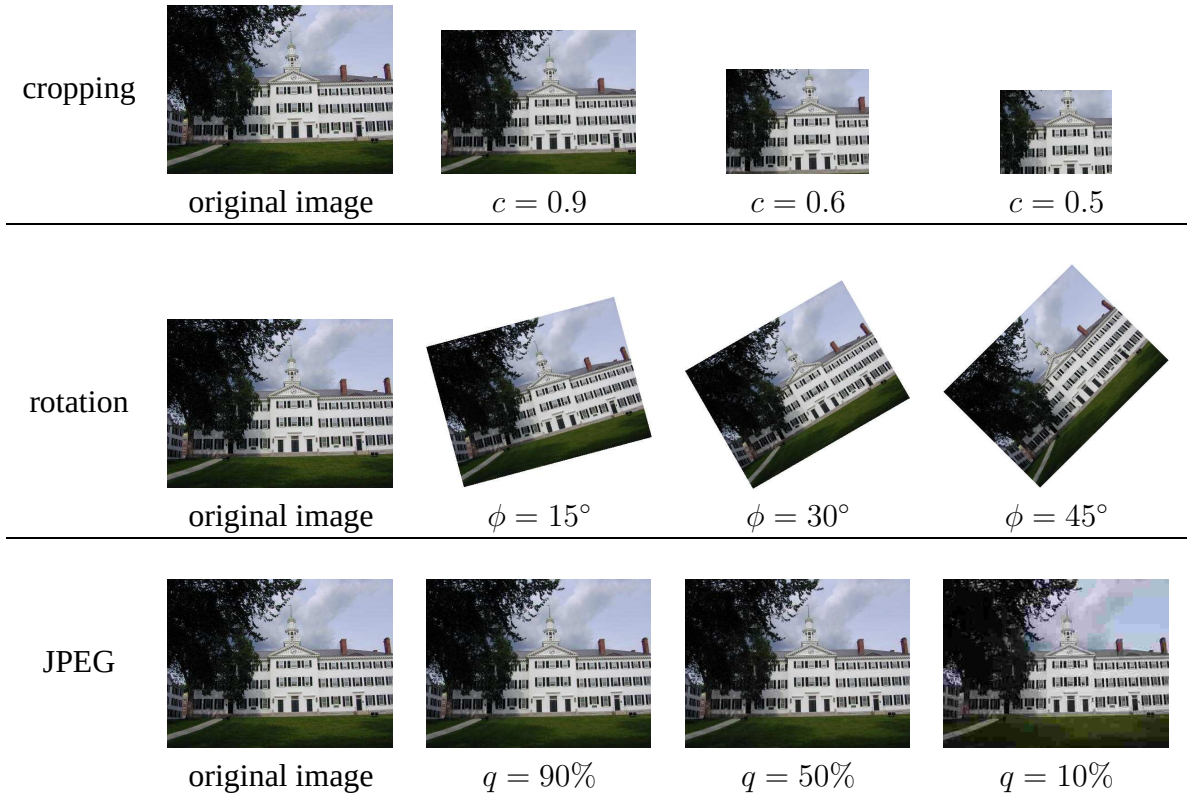


Figure 4.1: Examples of manipulated images for cropping, rotation, and JPEG compression.

classifier. This reflects the instationary nature of natural images in the spatial domain: the image statistics of the whole image can be quite different from those of a local region in the image.

For image rotation, it is interesting to note that the distance of image statistical feature vectors, starting with positive values, descends to negative as the rotation angle increased to 45°. Then the distance increases again to positive as the rotation angles continue to increase to 90°. This phenomenon, contrary to our initial assumption, is not caused by the interpolation nature of image rotation operation. We tested the one-class SVM classifier on images rotated with 15° and then -15° back. If the interpolation is the cause of this phenomenon, we should see a large difference between the resulted distances. However, what we observed, however, is that the relative difference between the one-class SVM classification results of the original images and the double rotated images are less than 3%. A more possible cause of the changes in the one-class SVM classification for the rotated images is that rotation changes the distribution of the local orientation energy in a natural image, thus affects the QMF coefficient marginal statistics. For instance, in a 45° rotated image, most of the horizontal and vertical energy in the original image will collapse into the diagonal subband, which is abnormal for natural images in the training set. On the other hand, though the role of vertical and horizontal statistics swaps in a 90° rotated image, it still keeps similar statistical regularities as the natural images in the training set.

Different JPEG quality also affects the image statistics and the one-class SVM classification. The major effect of JPEG compression on a natural image is the suppress of high frequency components, the degree of which is decided by the JPEG quality factor: the lower the quality is, the more high frequency components are lost in the compressed image. With a JPEG compress of quality 10, the compressed image has only one fiftieth of the size in bytes as the original image, but artifacts due to the loss of high frequency components, as well as the "blockiness" due to the segmentation of the JPEG process, are also clearly visible. As the first few scales in the wavelet and LAHD decomposition are mostly affected by the high frequencies in the image, the degradation affect them most. The one-class SVM classification captures this degradation of naturalness in JPEG images with decreasing qualities.

For additive noise, the one-class SVM classifier predicts a decrease of naturalness with higher noise level (lower SNR values), as they deviate from the training ensemble. Similar cases are true for blurring and phase perturbation.

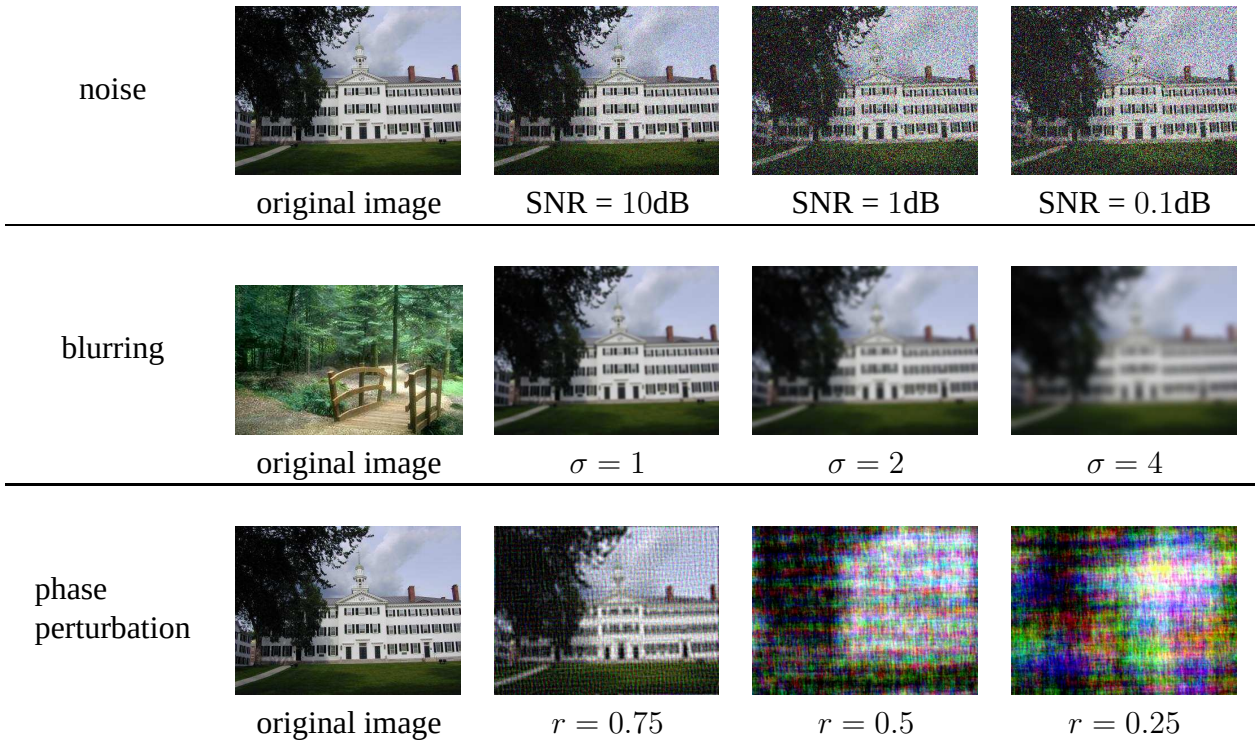


Figure 4.2: Examples of manipulated images for additive noise, blurring and phase perturbation.

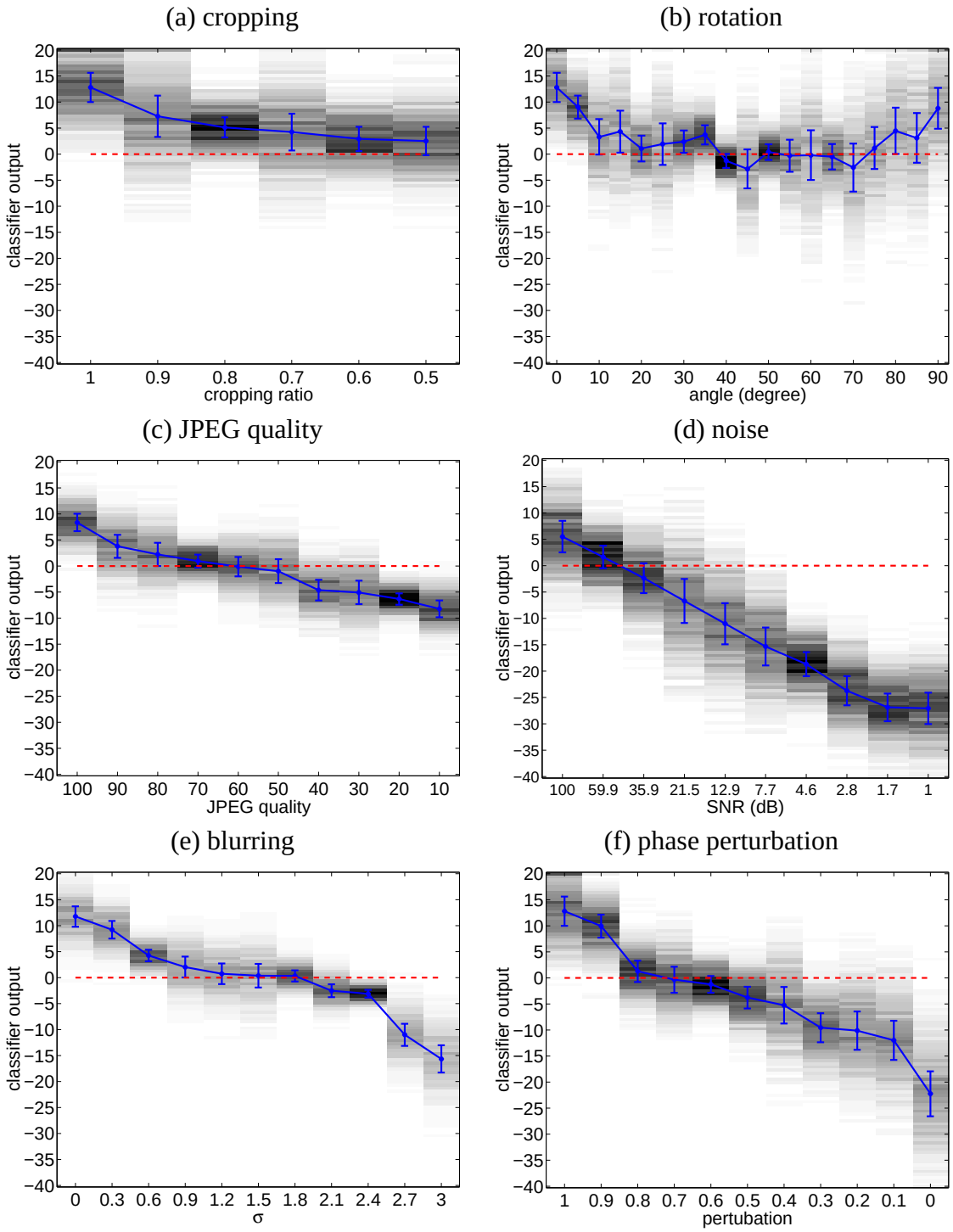


Figure 4.3: One-class SVM classifi cation of manipulated images. The dotted line indicates the hard classification boundary where negative values mean "unnatural". Shown on the curves are the mean values with the bar above and below the standard deviation. The histograms are shown as shaded blocks in background, with a white bin being 0 probability.

Chapter 5

Photographic vs. Photorealistic

In this chapter, the proposed image statistics introduced in the previous chapters are combined with non-linear classification techniques and applied to the task of differentiating natural photographic images and computer-graphics generated photorealistic images. We also compared the learned classification system with the performance of human subjects in a series of psychophysical experiments.

5.1 Introduction

In an age prevailing with digital media, it is no longer true that seeing is believing. This is partly attributed to the development of sophisticated computer graphics rendering algorithms [18] (e.g., ray-tracing, radiosity and photon-mapping) and softwares that can generate remarkably photorealistic images. These technologies have started challenging our long-held notion of photorealism. For instance, it is not easy to tell, from the two images in Figure 5.1, which one is a photograph and which one is rendered with a computer program.

Somehow unexpectedly, this technology also bears legal implications. In 1996, the United States Congress passed *The Child Pornography Prevention Act*, which in part prohibited any image that *appears to be* or *conveys the impression of* someone under 18 engaged in sexually explicit conduct. This law made illegal the computer-generated images that only appear to show minors involved in sexual activity. In 2002, however, the United States Supreme Court struck down portions of this law in their 6-3 ruling in *Ashcroft v. Free Speech Coalition* - the court said that the language in the 1996 *The Child Pornography Prevention Act* was unconstitutionally vague and far-reaching. This ruling essentially legalized the computer-generated child pornographic images and makes it considerably more difficult for law-enforcement agencies to prosecute such crimes - anyone trafficking these illegal images may always claim that they are computer generated to avoid punishment. Therefore, the law-enforcement agencies are in a great need of methods that can reliably differentiate between true photographic images and computer-generated photorealistic (hereafter, photorealistic) images.

One promising methodology to solve this problem is to take the advantage of the statistical regularities in natural photographic images. No matter how visually resembling a photographic

Figure 5.1: Examples of highly convincing photorealistic image generated by computer graphics software. The image in the left panel is a photograph while the one on the right is created by a 3D rendering software (3D Max Studio). Images from `www.fakeorfoto.com`.

image, a photorealistic image is created from a fundamentally different process. A photographic image is the result of the physical world projected on the image sensors in imaging devices, such as the film in an optical camera or the CCD (charge-coupled device) in a digital camera. On the other hand, photorealistic images are produced by rendering algorithms that simulate this imaging process. The rendering algorithms can only roughly model the highly complex and subtle interactions between the physical world and the imaging device. Such discrepancy will well reveal themselves in image statistics. This is the intuitive motivation of applying the proposed image statistics (Chapter 2) for this task. Differentiating photographic and photorealistic images then proceeds as a binary classification problem, where the type of an unknown image is automatically determined based on the proposed image statistics.

Previous Work

Though techniques able to generate highly convincingly photorealistic images have existed for more than two decades, relatively few computational techniques exist to differentiate between photographic and photorealistic images. There has been some work in evaluating the photorealism of computer graphics rendered images from the human perception point of view (e.g., [48, 47, 61]), with the aim to help improving the modeling and rendering algorithms to achieve higher degree of photorealism. In computer vision and pattern recognition, there are also some related work, though not directly applicable, on using statistical image features to differentiate or classify different classes of images. These works include techniques to differentiate between photographic and (non-realistic) graphical icons [3], city and landscape images [77, 74], in-door and out-door images [71], photographs and paintings [13], content-based image retrieval [5], texture classification [27], and scene identification [73].

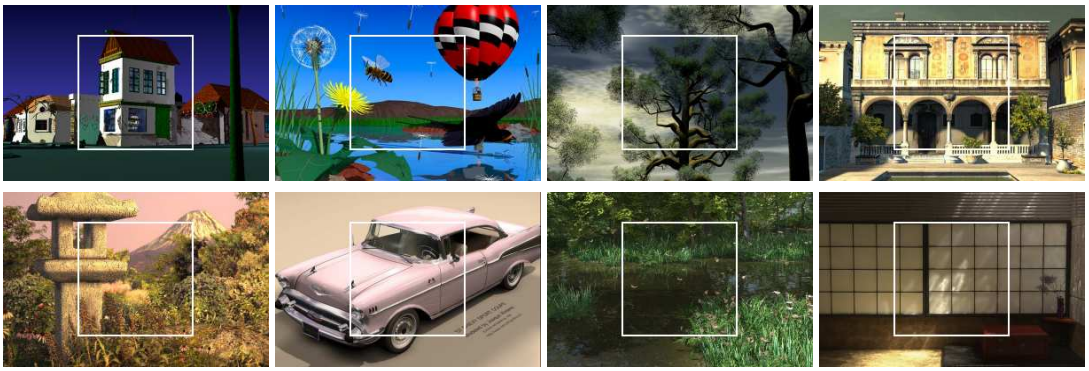


Figure 5.2: Eight examples from the $6,000$ photorealistic images. The white boxes are the central 256×256 region of the image from which the image statistics are measured. Note the variance in the image contents and levels of photorealism.

5.2 Experiments

We formulate the problem of differentiating photographic and photorealistic images as a binary classification, where a classifier is trained to determine if an image is photographic or photorealistic. In building the classifier, the training data are essential, as we want to avoid learning accidental difference between photorealistic and photographic images in color, texture, or other aspects of image contents. To this end, we need to train the classifiers on a considerably large set of images with contents as diverse as possible, so as to integrate out superficial difference in image contents. For the photographic images, we used the $40,000$ natural images as described in Chapter 1. For the photorealistic images, $6,000$ were downloaded from `www.raph.com` and `www.irtc.org`. Shown in Figure 5.2 are eight samples from the $6,000$ photorealistic images. The relative fewer number of photorealistic images reflects the fact that photorealistic images, especially those of high quality, require more effort to create. All photorealistic images are color (RGB), JPEG compressed (with an average quality of 90%), and typically on the order of 600×400 pixels in size. Visually, these photorealistic images span a range of contents (e.g., landscapes and city scenes) and imaging conditions (e.g., indoor and outdoor lighting, close up and far away views, etc.), and have different levels of photorealism. They were created from popular computer graphics software packages (e.g., 3D Studio Max, Maya, SoftImage 3D, PovRay, Lightwave 3D and Imagine).

From the $40,000$ photographic images and $6,000$ photorealistic images, $32,000$ photographic and $4,800$ photorealistic images were randomly chosen to form the training set for both an LDA and a non-linear SVM classifier[1]. From each image, training and testing alike, the proposed image statistics (Chapter 2) were extracted. Specifically, six types of statistical features can be formed from these image statistics, as

1. 72-D feature vector of grayscale local magnitude statistics;

2. 36-D feature vector of grayscale local phase statistics;

[1]The SVM algorithm was implemented with the package LIBSVM [9].

3. 108-D feature vector of grayscale local magnitude and local phase statistics;

4. 216-D feature vector of color local magnitude statistics;

5. 216-D feature vector of color local phase statistics;

6. 432-D feature vector of color local magnitude and local phase statistics.

To accommodate different image sizes, only the central 256×256 region of each image was analyzed. During the training phase, the false-negative rate (i.e., the probability of a photorealistic image being classified as a photographic image) was controlled to be less than 1%. This specific setting reflects the requirement in practice, where the cost of a false negative is much higher than a false positive, and also sets the comparisons henceforth described on a fair ground.

Grayscale or Color

We first investigate whether the statistical regularities among different color channels are important in differentiating photographic and photorealistic images. Shown in Figure 5.3 are the classification performance of the LDA classifiers with a training false negative rate of 1% for, (a) 108 grayscale image statistics (72 local magnitude statistics and 36 local phase statistics) and (b) 432 color image statistics (216 local magnitude statistics and 216 local phase statistics). The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set. For the sake of comparison, the results for the color image statistics in panel (b) are annotated with the results of the grayscale image statistics, panel (a). To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images. On average, for a 1.1% false negative rate, the grayscale image statistics afford a 21.2% accuracy on the photographic images, with a 4.2% standard deviation over the 100 random training/testing splits. For the color image statistics, the accuracy on photographic images is 54.6% on average, with a 7.8% standard deviation and a 1.2% false negative rate. The color image statistics clearly outperformed the grayscale image statistic. Also, note that, in both experiments, the testing performance was fairly close to the training performance, indicating that none of classifiers overfit the training data. Also, the testing false negative rates, obtained by subtracting from the classification accuracies of the photorealistic images, were consistent with the settings of less than 1%.

Linear or Non-linear Classification

We next compared the performance of different classification techniques. Specifically, we show, in Figure 5.4(a), the classification accuracies of a non-linear SVM classifier with RBF kernel, on the 432 color image statistics and a 1% false negative rate. The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set. For the sake of comparison, the results for the non-linear SVM classifier are annotated with the results of the LDA classifier, Figure 5.3(b). To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the

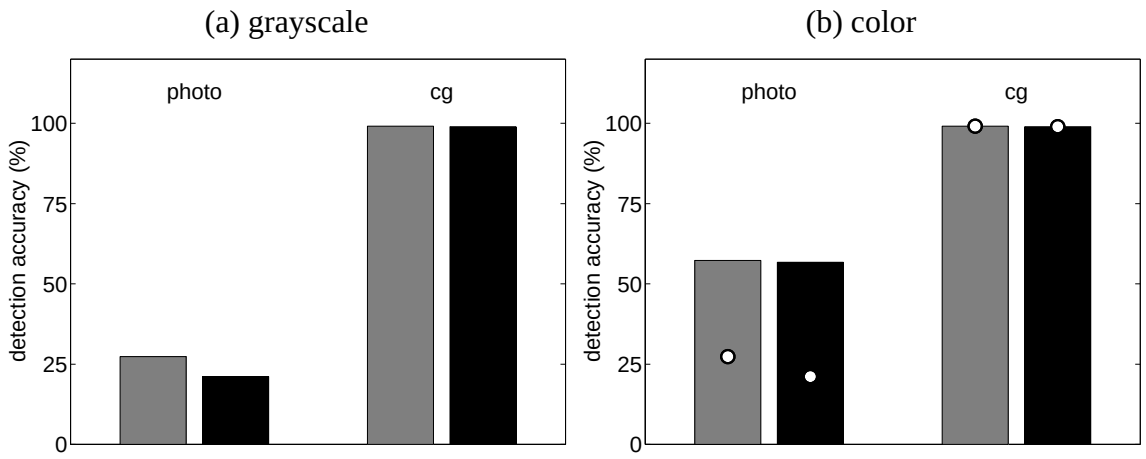


Figure 5.3: Classification accuracies of LDA classifiers with (a) 108 grayscale image statistics (72 local magnitude statistics and 36 local phase statistics) and (b) 432 color image statistics (216 local magnitude statistics and 216 local phase statistics). The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set, for photographic (photo) and photorealistic (cg) images. For the sake of comparison, the results for the color image statistics in panel (b) are annotated with the results of the grayscale image statistics, panel (a). To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images.

$46,000$ images. On average, the non-linear SVM classifier affords a 74.3% accuracy on the photographic images, with a 10.1% standard deviation over the 100 random training/testing splits. In general, the non-linear SVM generally outperformed the linear LDA classifier, though with a more complicated training process. This suggests that a non-linear separating surface represented by a non-linear SVM classifier better describes the difference in image statistics between photographic and photographic images.

False Negative Rates

We also investigate the sensitivity of the classification to different false negative rates, or the error rate of misclassifying a photorealistic image as a photographic image. In general, the false negative rate and the classification accuracy of photographic images are positively correlated: the false negative rate's increase implies an increase in the classification accuracy of photographic images. Shown in Figure 5.4(b) are the classification accuracy of a non-linear SVM classifier with the 432 color image statistics and a 0.5% training false negative rate. The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set. For the sake of comparison, these results are annotated with the results of the non-linear SVM classifier with a 1% training false negative rate, Figure 5.4(a). To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images. On average, the classification accuracy of photographic images are 60.4%, with a 8.4% standard deviation. A small change in false negative

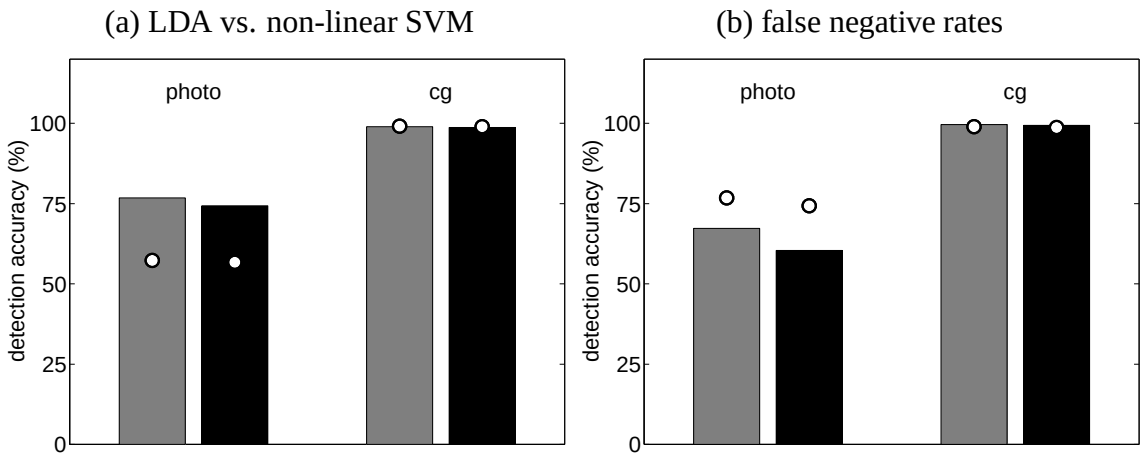


Figure 5.4: Classification accuracies of non-linear SVM classifiers with the 432 color image statistics, for (a) a training 1% false negative rate and (b) a training 0.5% false negative rate. The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set, for photographic (photo) and photorealistic (cg) images. For the sake of comparison, the results in panel (a) are annotated with the results in Figure 5.4(b) and the results in panel (b) are annotated with the results in panel (a). To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images.

rate (0.5%) results in a relatively large change in the classification accuracy (about 14%).

Categories of Image Statistics

For classifiers with high-dimensional image features, it is a natural question whether we need all the components, in other words, is there a minimum set of statistics that perform as well. A class-blind global dimensionality reduction of the high-dimensional feature (e.g., PCA) is inappropriate, as the class-specific information is required in classification. Furthermore, we would also like to know which category of the image statistics, i.e., local magnitude statistics or local phase statistics, and within local magnitude statistics, marginal statistics or linear prediction error statistics, had more contribution in the final classification. This knowledge will further justify our choice of the image statistics.

Shown in Figure 5.5, from left to right, is the detection accuracy for a non-linear SVM trained with the 108 color coefficient marginal statistics only, the 108 magnitude linear prediction error statistics only, the 216 local phase statistics only, and the 216 local magnitude statistics including both coefficient marginal and magnitude linear prediction error statistics. For point of comparison, the dots correspond to a non-linear SVM trained on the complete set of 432 color image statistics with both the 216 local magnitude and the 216 local phase statistics. These results show that the combined local magnitude and local phase statistics provide for better classification accuracy than only a subset of the statistics in differentiating photographic and photorealistic images.

To get a better picture of the role played by individual image statistics and statistics categories,

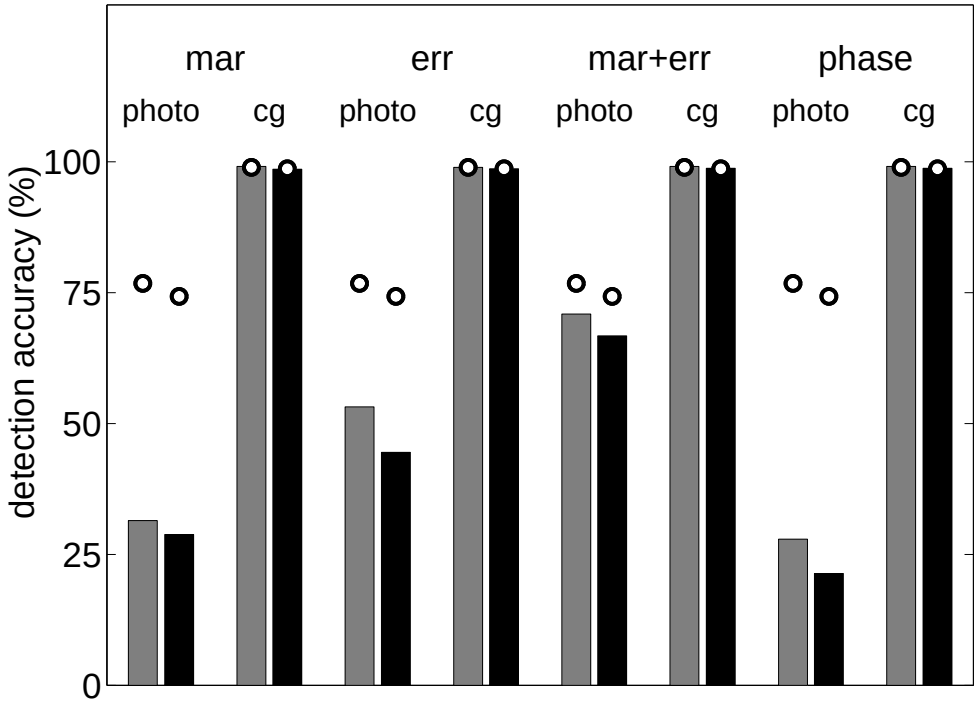


Figure 5.5: Classification accuracy for a non-linear SVM trained on (from left to right) the 108 color coefficient marginal statistics only, the 108 magnitude linear prediction error statistics only, the 216 local phase statistics only, and the 216 local magnitude statistics including both coefficient marginal and magnitude linear prediction error statistics. The dots correspond to a non-linear SVM trained on the complete set of 432 color image statistics with both the local magnitude and local phase statistics. The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set, for photographic (photo) and photorealistic (cg) images. To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images.

we performed experiments with LDA classifiers[2] with individual statistics being incrementally included. We tested on both the 216 local magnitude statistics and the 432 local magnitude and local phase statistics. Specifically, for the 216 local magnitude statistics, we began by choosing the single statistics, out of the 216 possible coefficient marginal and magnitude linear prediction error statistics, that has the best classification accuracy on photographic images, while keeping a less than 1% false negative rate. This was done by building 216 individual LDA classifiers on each statistics, and choosing the one that yielded the highest accuracy (the feature was the variance in the error of the green channel's diagonal band at the second scale). The next best feature was then chosen from the remaining 215 statistics. This process was repeated until all 216 statistics were

[2]This analysis was performed only on the LDA because the computational cost of retraining $23,220 = 216 + \cdots + 1$ or $93,528 = 432 + \cdots + 1$ non-linear SVMs is prohibitive. We expect the same pattern of results for the non-linear SVM.

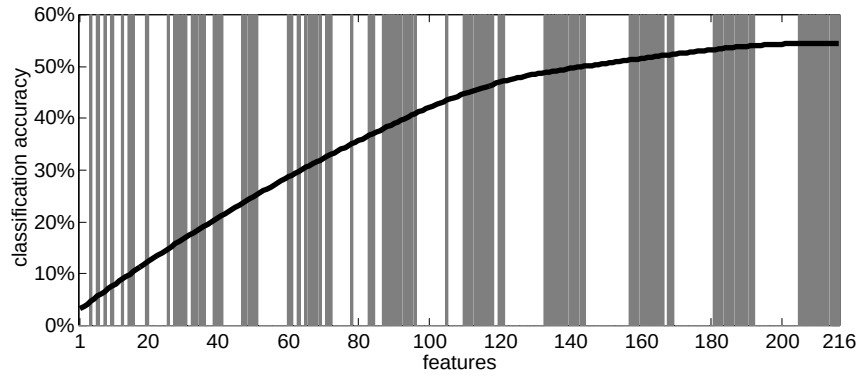


Figure 5.6: Classification accuracy of LDA classifiers as a function of the number and category of the 216 local magnitude statistics with a 1% training false negative rate for photorealistic images. The horizontal axis corresponds to the number of statistics incorporated, and the vertical axis corresponds to the detection accuracy in percentage. The white and gray stripes correspond to magnitude linear prediction error and coefficient marginal statistics, respectively.

selected.

Shown in Figure 5.6 is the classification accuracy (solid line) plotted as a function of the number and category of statistics for the LDA classifier. The white and gray stripes correspond to magnitude linear prediction error and coefficient marginal statistics, respectively. If the statistics included in the i^{th} iteration is of coefficient marginal, then at the i^{th} position on the horizontal axis a vertical gray line is annotated, and if the statistics is of magnitude linear prediction error, a vertical white line is drawn at the i^{th} position. The shape of the solid curve reflects a decreasing contribution to classification of later incorporated statistics. The interleaving pattern of the coefficient and error statistics suggests that both types of statistics are important for classification. Also, it was observed that the last few statistics being included are the means and skewness of both types of statistics, which are mostly close to zero and thus carry less useful information for classification.

A similar experiment was also performed on all of the 432 image statistics, including the 216 local magnitude and the 216 local phase statistics. Similar to the previous experiment, we built LDA classifiers by incrementally include components from the 432 statistics and studied the occurrence pattern of both types of statistics. As shown in Figure 5.7, an interleaving pattern is observed for decomposition and local phase statistics, indicating that they are both important for the classification.

Permutation Test

One concern of using a complicated non-linear classification technique such as a non-linear SVM is that it may be "too powerful" for the classification task, i.e., it can learn arbitrary labeling of the training data. This should be avoided as the learned classifiers do not reflect fundamental difference in the image statistics between photographic and photorealistic images. To confirm that

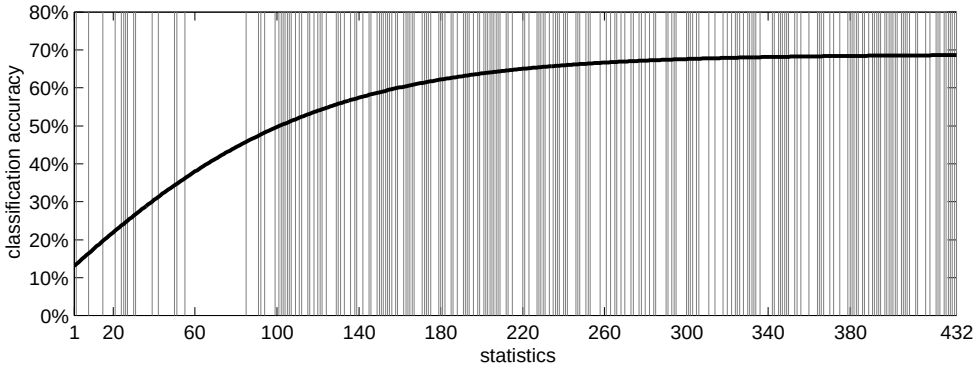


Figure 5.7: Classification accuracy of LDA classifiers as a function of the number and category of the 432 local magnitude and local phase statistics with a 1% training false negative rate for photorealistic images. The horizontal axis corresponds to the number of statistics incorporated, and the vertical axis corresponds to the detection accuracy in percentage. The white and gray stripes correspond to decomposition and local phase statistics, respectively.

non-linear SVM is appropriate for this application, we trained non-linear SVM classifiers with the 432 color image statistics and a 1% training false negative rate, with random class labels assigned to the training and testing images (this practice is commonly referred in pattern recognition as the permutation test [64]). We expect a random class assignment to lead to significantly worse classification accuracy. Specifically, we generated ten different training/testing splits from the $46,000$ photographic and photorealistic images with their class labels randomized. Ten different non-linear SVM classifiers were then trained and tested on these randomly shuffled class labels. The best performance across the ten training sets was 32.1% for correct classification of the photographic images, with a 1.4% false-negative rate. Note that this is significantly worse than the 74.3% detection accuracy obtained when the correct training labels were used. Therefore, the use of non-linear SVM to differentiate photographic and photorealistic images is justified, as they are not able to learn random labellings from the training data.

5.3 Comparison with Other Feature Types

Experiments described in previous sections empirically justified the use of the proposed image statistics for differentiating between photographic and photorealistic images. In this section, we compare their performance with that of other image statistics proposed recently in computer vision and pattern recognition. A majority of these works are based on multi-scale image decompositions (e.g., wavelets) that decompose an image into basis localized in both spatial and frequency domains. As shown in Chapter 1, such an image decompositions better capture statistical regularities in natural images than the representations based on pixel intensities or a global Fourier transform. Specifically, in this section, we compared the performance of three statistical image features, namely, multi-scale thumbnails [5], multi-scale histograms [27] and gists [73], with that of the proposed image statistics on differentiating photographic and photorealistic images. One

word of caution is that the comparison is innately unfair, in that all compared image features are not originally designed for the purpose of differentiating between photographic and photorealistic images, and their inferior performance on this task will not discount their usefulness in other applications.

Multi-scale Thumbnails

The multi-scale thumbnail [5] is a multi-dimensional statistical image feature formed from the outputs of a series of linear filters that capture local energy across spatial locations, orientations and different scales. These filters form an over-complete frame of image which are similar to those from a wavelet decomposition. The multi-scale thumbnails were originally designed for content-based image retrieval, where images in a large database were reduced to thumbnails for comparison and retrieval. Specifically, the multi-scale thumbnails are formed as following: at the highest scale of resolution, the image is convolved with a set of N linear filters. The resulting N filtered images are then rectified by squaring the corresponding filter outputs, which measure the local energy in the filter responses. Next, the rectified filter-response images are downsampled by a factor of two, and then they are used as inputs for the N filters in another round of iteration. With L levels of processing, this process yields N^L output images at completion. The feature vector of the original image is then formed by the means of these N^L images, which are the squared magnitudes of each subband. For RGB color images, the number of statistics is tripled to $3N^L$. In our implementation, a filter bank with 6 filters consisted of Gaussian and its first and second order derivatives with different orientations was employed, and three levels of analysis were performed, which resulted in a feature vector of 648 dimensions.

Multi-scale Histogram Features

In [27], a statistical image features based on image histograms in multiple scales is proposed for texture classification. Using the whole histograms (sampled probability density functions) as image features has been popular in computer vision, as in object recognition [39] and image retrieval [8]. Compared to individual statistics, histograms may carry more information. Specifically, the multi-scale histogram image features are formed by the histograms of each scale of an L scale Gaussian pyramid decomposition. Specifically, the histogram for the l^{th} scale is obtained as a result of binning the whole decomposition scale with $B(1/2)^{(L-l)}$ bins, yielding a feature vector of dimension $B(2-(1/2))^L$. A RGB color image will have a feature vector of dimension $3B(2-(1/2))^L$. In our implementation, we build a four-level Gaussian pyramid on each color channel, and with the base bin number of 80, resulting in a feature vector of dimension 450 across different scale and color channels.

Gists

In [73], an image statistical feature termed as "gist"' is used for object recognition and scene categorization, which is a multi-dimensional feature vector collected from a multi-scale image decomposition. Specifically, on each color channel of an image, a four-level six-orientation steerable

pyramid [69] is constructed. For each subband in the steerable pyramid, the coefficient magnitudes are raised to two and four order to capture the second and higher order energy in the subband. These power-raised magnitudes in each subband are further spatially averaged into 4×4 blocks, yielding a $768 = 2 \times 4 \times 4 \times 4 \times 6$ dimensional feature vector. To reduce the dimensionality and effects of noise, PCA is performed on the these 768-D vectors from a large set of images. The projections on the top 160 principal components are preserved for each color channel and stacked as the final feature vector of 480 dimensions.

Experiments

To empirically compare the multi-scale thumbnail, multi-scale histogram and gist features with the image features based on the proposed image statistics on differentiating photographic and photorealistic images, we performed similar experiments as in the previous section. Specifically, we randomly split the $40,000$ photographic and $6,000$ photorealistic images into one training set of $32,000$ photographic and $4,800$ photorealistic images and one testing set of $8,000$ photographic and $1,200$ photorealistic images. From each image, training and testing alike, the multi-scale thumbnail, multi-scale histogram and gist features were collected from the central 256×256 region to accommodate the different image sizes. Then non-linear SVM classifiers were trained based on these image features on the training set. In accordance to the experiments in section 5.2, all non-linear SVM classifiers were controlled to have a 1% training false negative rate (probability of classifying a photorealistic image as photographic).

Shown in Figure 5.8 are classification accuracy for a non-linear SVM trained on the 648-D multi-scale thumbnail features (thmb), 450-D multi-scale histogram features (hist) and 480-D gist features (gist) with a 1% training false negative rate. The dots correspond to a non-linear SVM trained on the 432 color image statistics. The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set, for photographic (photo) and photorealistic (cg) images. To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images. With around 1% false negative rate, the multi-scale thumbnail feature achieved an average classification accuracy of 29.8% on photographic images, with a standard deviation of 5.2%; the multi-scale histogram features had a classification accuracy of 47.1% with a standard deviation of 7.8%; and the gist features had a classification accuracy of 48.3% with a 8.4% standard deviation. This performance are clearly less competitive to that of the image features based on the proposed 432 color image statistics. Besides the fact that these image features were not originally designed for differentiating between photographic and photorealistic images, another important reason is that for all these image features, higher-order correlations within a multi-scale image decomposition are not modeled, which our previous experiments confirmed to have an important role.

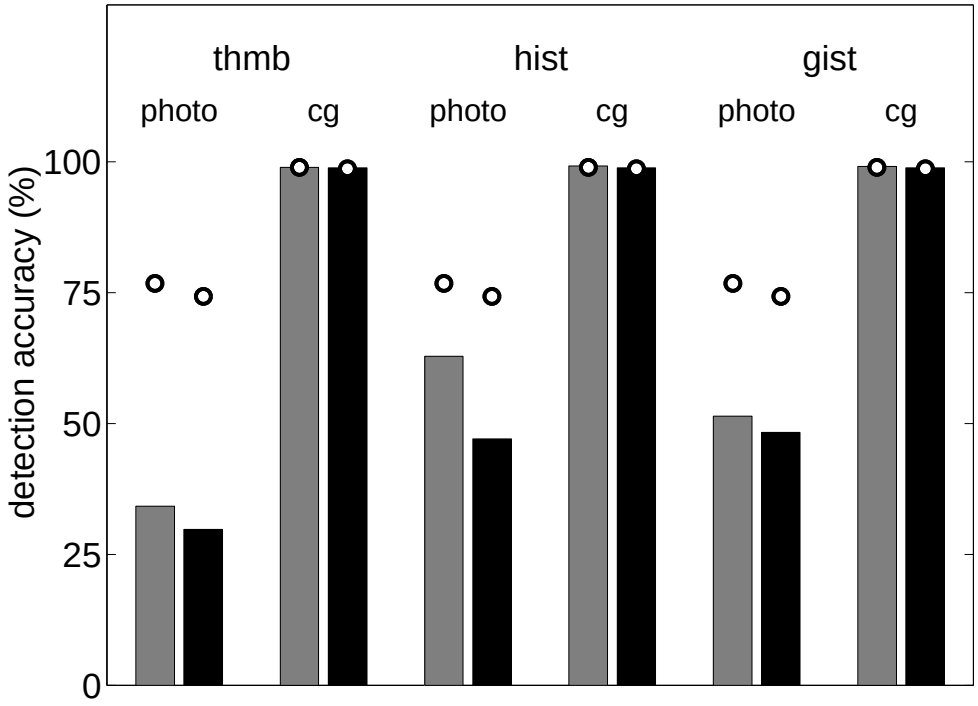


Figure 5.8: Classification accuracy for a non-linear SVM trained on the 648-D multi-scale thumbnail features (thmb), 450-D multi-scale histogram features (hist) and 480-D gist features (gist) with a 1% training false negative rate. The dots correspond to a non-linear SVM trained on the 432 color image statistics. The gray bars correspond to the accuracies on the training set and the black bars correspond to the accuracies on the testing set, for photographic (photo) and photorealistic (cg) images. To avoid reporting performance of a specific training/testing split, what is reported is the classification accuracy averaged over 100 random training/testing splits of the $46,000$ images.

5.4 Visual Relevance

Human vision system (HVS) is inarguably the ultimate performance benchmark of a computer vision or pattern recognition system. In our case, we would like to investigate the similarity and difference of our image statistics based classification systems of photographic and photorealistic images with the HVS, and compare their practical performance. Empirically comparing with the performance of the HVS not only provides a more meaningful evaluation of our method, but also may shed light on the role played by the proposed image statistics in this task.

We started with the investigation of the visual relevance of the classification results of the nonlinear SVM classifier and the 432 color image statistics[3]. To this end, we trained another non-linear SVM classifier on the training set as described in Section 5.2. However, we enforced no constraints on the false negative rate, as we aimed to find the optimal classification surface

[3]Similar comparison can also be made for other types of image statistics and classifiers. The current choice is for its best practical performance.

Figure 5.9: Eight examples of correctly classified photographic images.

between the image statistics of photographic and photorealistic images without biasing to one type of error rate. The non-linear SVM classifier, in the testing stage, achieved a 85.4% classification accuracy for photographic images, and a 79.3% classification accuracy for photorealistic images, which yields an overall accuracy of 84.6%.

Next, we inspected the classification results on individual images to investigate the visual relevance of the classification results. Particularly, we were interested in images that were correctly or incorrectly classified by the SVM classifier. Shown in Figures 5.9 are eight examples of correctly classified photographic images, and in Figure 5.10 are eight examples of correctly classified photorealistic images. Many of the correctly classified photographic images are of typical scenes from natural environment, coinciding with our intuitive notion of natural photographic images. However, some man-made objects (e.g., the high-way and trucks) were also correctly classified. Most of the photorealistic images assume some particular artificial appearance (e.g., a frog with plastic skin), and thus will not be challenging for human viewers. It was the incorrectly classified images that were more interesting, as they shed lights on whether the image statistics correlate with the image contents. Shown in Figure 5.11 are eight examples of incorrectly classified photographic images, and in Figure 5.12 are eight examples of incorrectly classified photorealistic images. Interestingly, many of the incorrectly classified photographic images are photographic images with low photorealism, such as the images of road signs and posters. The 2-D nature of the objects in the image makes these images easy to be created with algorithms. On the other hand, many incorrectly photorealistic images depict a vivid scene from natural environment, and are visually hard to differentiate from a photographic image. However, there are also incorrectly classified photorealistic images that are easy to discount visually.

Furthermore, we tested this non-linear SVM classifier on a novel set of fourteen images (7 photographic, 7 photorealistic) from the website `www.fakeorfoto.com`. These images were used to test the viewers' ability to differentiate photographic and photorealistic images, for purely amusement purpose. Of all the 14 images, the SVM classifier correctly classified 11 images, or a 78.6% overall classification accuracy, consistent with the results reported on our testing set. Shown in Figure 5.13 are the fourteen images with the correctly classified photographic images in the top row, and the correctly classified photorealistic images in the second row. Shown in the two bottom rows are one incorrectly classified photographic image (c) and two incorrectly classified

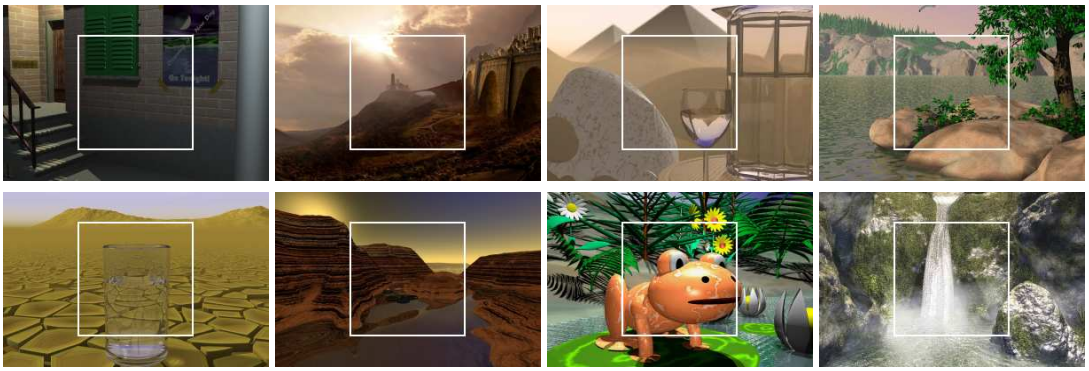


Figure 5.10: Eight examples of correctly classified photorealistic images.

Figure 5.11: Eight examples of incorrectly classified photographic images.

photorealistic images (d). The fact that the classifier achieved consistent performance on a novel set of images in neither the training nor the testing set confirms that the the trained SVM classifier can generalize to novel data.

Psychophysical Experiments

To be able to compare with the performance of the HVS precisely, a series of psychophysical experiments were conducted[4].

Stimuli: The stimuli were 111 landscape photographic and 111 landscape photorealistic images not included in the $46,000$ images used to train and test classifiers in our previous experiments. These images were hand-picked to ensure that it is not possible to classify them from the contents. The stimuli were then displayed on an Apple PowerBook. The viewing distance was not constrained but was typically about 50 cm. The sizes of the stimuli were 600×400 pixels, corresponding to 15×10 cms in dimension.

Observers: 22 observers were recruited from the introductory psychology subject pool at Rutgers Camden campus. The only restriction on participation was normal or corrected-to-normal acuity

[4]These experiments and the following description were conducted and provided by Professor Mary Bravo of the Psychology Department at the Rutgers University, Camden, NJ.

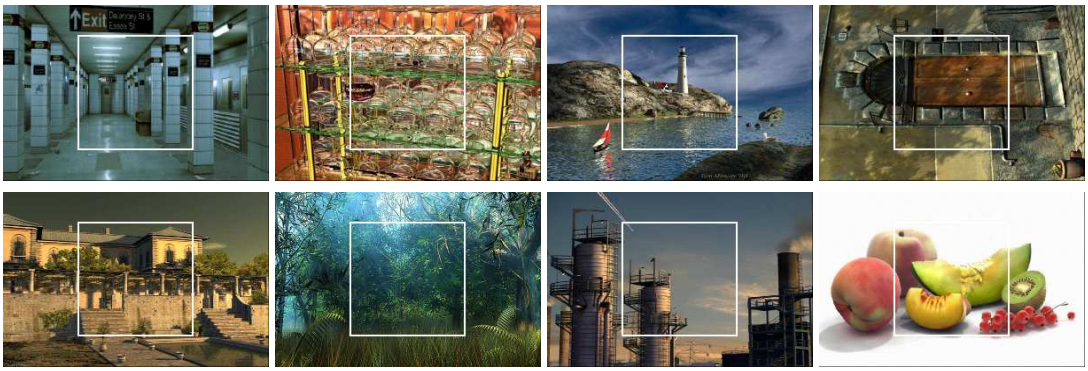


Figure 5.12: Eight examples of incorrectly classified photorealistic images.

and normal color vision. None of the observers had previous experience in a psychophysical experiment.

Procedure: Each stimulus was displayed for one second. The observer's task was to judge whether the image was a photographic or a computer generated scene. Judgments were indicated by pressing one of two buttons on the computer keyboard. After the observer pressed a key, there was a one second delay before the next stimulus was presented. No feedback was given. The 222 stimuli were divided into four blocks of 44 trials and one block of 46 trials. In each block there was an equal number of photographic and photorealistic stimuli. The order of the blocks was randomized across observers.

The mean performance on photographic images of the 22 observers is 80.6% with a standard deviation of 11.3%, with max/min values of 93.7% and 48.1%, respectively. The mean performance on photorealistic images of the observers is 81.7% with a standard deviation of 6.6%, with max/min values of 93.7% and 69.4%, respectively.

Computational Experiment

In accordance with the psychophysical experiments, 20 non-linear SVM classifiers were trained on the $40,000$ photographic images and $6,000$ photorealistic images and tested them on the 222 landscape photographic and photorealistic images used in the psychophysical experiments. To fairly compared with the corresponding psychophysical experiments, each classifier was trained on partly overlapped data, so to reduce the variance in performance, well preserving some relative independence. Specifically, the $40,000$ photographic and $6,000$ photorealistic training images were first divided into a shared group of $2,000$ images and other 20 equally divided groups of size $2,200$. The training set of each classifier was the combination of the shared group with one of the 20 groups. The training set of each SVM classifier had $4,200$ images, $3,650$ of which were of photographic and 550 were of photorealistic. Each SVM classifier, with RBF kernels, was trained on its dedicated training sets. The mean testing accuracy for photographic images is 80.0%, with a standard deviation of 4.4%. The maximum and minimum are 71.2% and 90.9%, respectively. The mean testing accuracy for photorealistic images is 84.8%, with a standard deviation of 12.3%. The maximum and minimum are 100.0% and 56.7%, respectively. Shown in Figure 5.14 are the mean

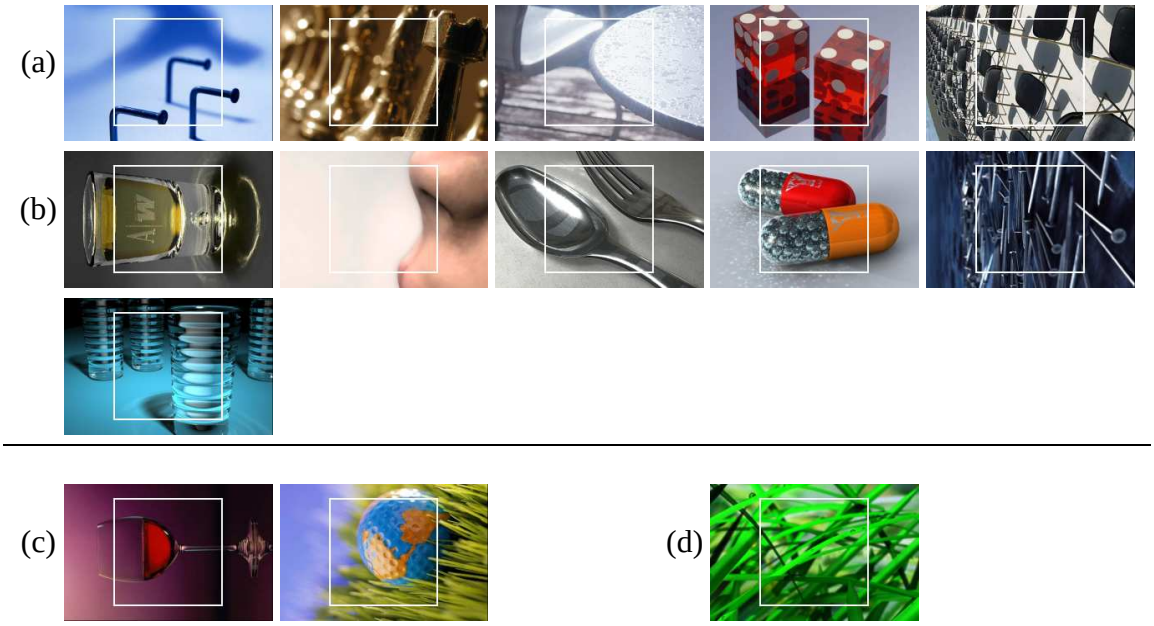


Figure 5.13: Images from `www.fakeorfoto.com`. Shown in (a) and (c) are correctly and incorrectly classified photographic images, respectively. Shown in (b) and (d) are correctly and incorrectly classified photorealistic images, respectively.

classification accuracies for the 22 human observers (gray bars) and 20 non-linear SVMs (black bars), on 111 photographic (photo) and 111 photorealistic (cg) images.

Comparison and Analysis

The performances of human observers and the non-linear SVM were first on the qualitative level. The class label of each image was first determined by a majority vote of all participating observers or SVM classifiers. For the 111 photographic images, voting results of the human observers agreed on 102 (91.9%) with those of the SVM classifiers , with 101 images being correctly classified and 1 images being incorrectly classified by both. On the 9 images they did not agree, 7 were classified as photographic by human observers and photorealistic by our SVM classifiers, and 2 were classified as photorealistic by human observers and the reverse by the SVM classifiers. For the 111 photorealistic images, the human observers and the SVM classifier agreed with each other on 96 (86.5%) images, with 92 being correctly classified and 4 being incorrectly classified. The SVM classifier incorrectly classified 7 photorealistic images which were correctly classified by human observers, and 8 other images were in the opposite case. This suggests an overall consistency between the human observers and the SVM classifiers.

We further compared quantitatively the classification results of human observers and SVM classifiers. We recorded the "yes/no" answers to each image of each observer on each image. For each image, the percentage of observers/SVM classifiers with correct classification was used

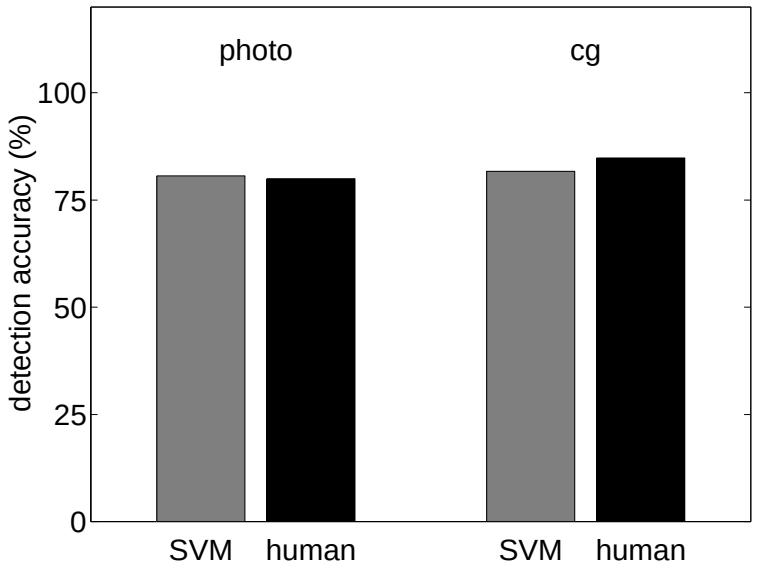
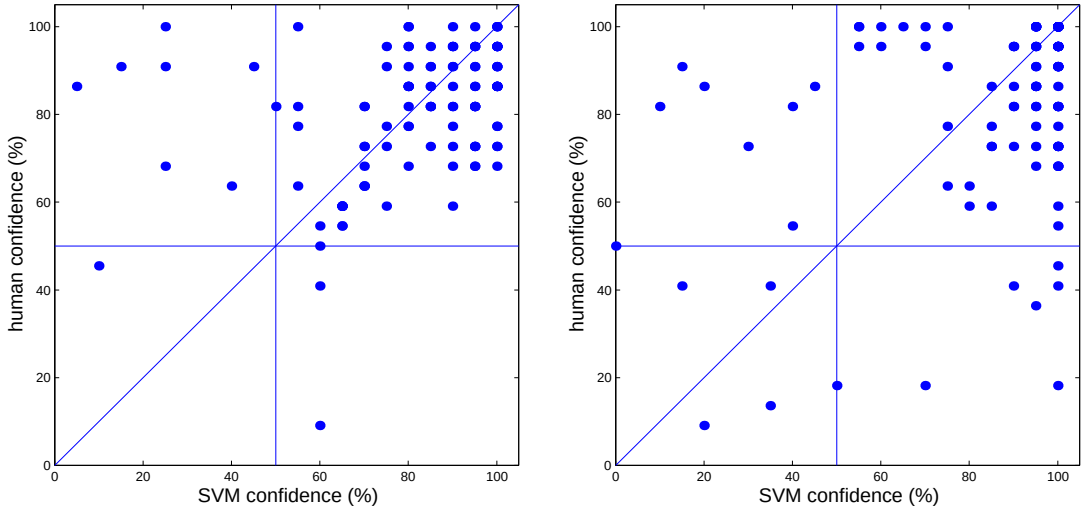


Figure 5.14: Mean classification accuracies for the 22 human observers (gray bars) and 20 non-linear SVMs (black bars), on 111 photographic (photo) and 111 photorealistic (cg) images.

as a confidence measure of classification. Shown in Figure 5.15 is the scatter plot of these confidence measures of the SVM classifiers (x-axis) and the human observers (y-axis), for photographic images (left), and photorealistic images (right). Also shown are the corresponding correlation coefficients[5]. The correlation coefficients suggest that there is no simple linear dependency yet the responses are not totally uncorrelated. In conclusion, the comparative experiments of the performance of the HVS and SVM classification based on the proposed image statistics suggest that the two classification systems performed the task almost equally well in the testing condition. However, there is also profound difference in their underlying mechanisms, making the proposed image statistics complementary to human visual inspection in differentiating between photographic and photorealistic images.

[5]A correlation coefficient for two random variables X and Y are defined as $\rho = \frac{\mathcal{E}\{(X-\mathcal{E}\{X\})(Y-\mathcal{E}\{Y\})\}}{\sqrt{var(X)var(Y)}}$. A ρ-value of 1.0 meaning linear dependency and a ρ-value of 0.0 statistical uncorrelated.



photographic $\rho = 0.30$ photorealistic $\rho = 0.36$

Figure 5.15: Scatter plot of the responses from the SVM classifier (x-axis) and the human observers (y-axis) for photographic images (left), and photorealistic images (right). Also shown is the corresponding correlation coefficients, ρ, with a value of 1.0 meaning linear dependency and a value of 0.0 statistical uncorrelated. The values shown here mean their is no simple linear dependency yet the responses are not totally uncorrelated.

Chapter 6

Generic Image Steganalysis

In this chapter, a generic image steganalysis system based on the proposed image statistics and non-linear classification techniques is presented. We begin with a brief review of image steganography and steganalysis in Section 6.1, then describe in detail the generic image steganalysis system for JPEG, TIFF and GIF images in Section 6.2 and 6.3, respectively.

6.1 Image Steganography and Steganalysis

The word *steganography* finds its root in Greek, literally meaning "covered writing". The goal of steganography is to hide messages in an innocuous cover medium in an unobtrusive way, so as to evade inspection. The earliest historical record of using steganography dated back to the Romans, and it has been widely used ever since for military and intelligence purposes. A notable example of steganography is the following message sent by a German spy during the first world war:

> Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit.Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.

The steganographic message is hidden in the second letter of each word as,

> Pershing sails from NY June 1.

Traditional steganography methods include invisible inks, micro dots, character rearrangement, covert channel and spread spectrum communication [10]. With the populace of digital media, digital images, audios, and videos become ideal covers for steganography. Especially, steganography in digital images has received the most attention (see [35, 2, 33, 53] for general reviews), for their wide availability, easy accessibility and large data volume - an image of size 640×480 pixels and 256 grayscales can hide up to 3 kilobytes of data, large enough to contain the whole text of the *Declaration of Independence*.

One simple yet effective method to achieve such embedding capacity is least significant bits (LSB) insertion, where message bytes are embedded into the lower bits (i.e., the less significant

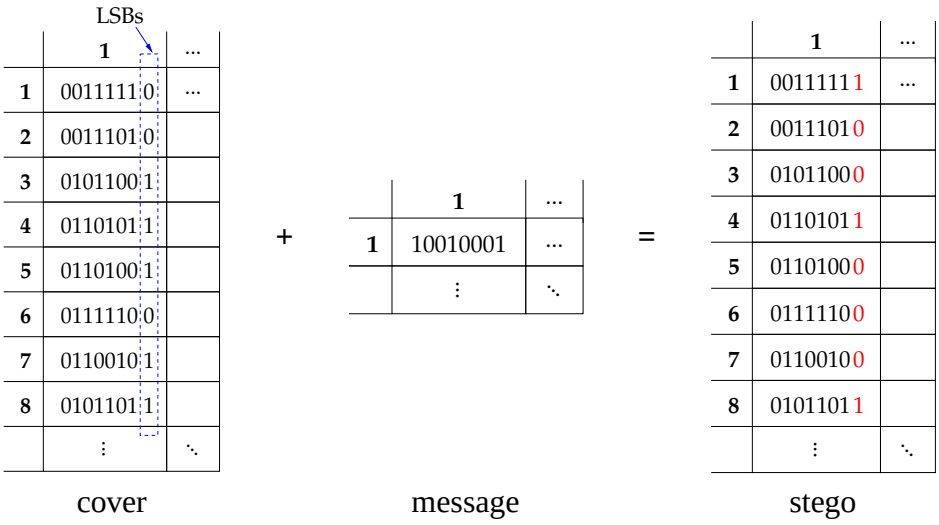


Figure 6.1: Least significant bit (LSB) insertion of a steganographic message in an 8-bit grayscale image (left column). One byte of the message (middle column) is embedded into the LSBs of 8 consecutive pixels in the cover image by coercing the eight LSBs of the eight pixels in the cover image to be the same as the byte in the message, (right column).

bits, usually the last bit which is the least significant bit) of the data bytes in the image file[1]. On uncompressed or lossless compressed image formats (e.g., TIFF and GIF), the LSBs of the raw pixel intensity are used for embedding. Shown in Figure 6.1 is a simple example of LSB insertion in an 8-bit grayscale image. One byte of the message, middle, is embedded into the LSBs of eight consecutive pixels in the cover image by coercing the eight LSBs of the eight pixels in the cover image to be the same as the byte in the message, right. In a similar fashion, an $M \times N$ 8-bit grayscale image can hide a message of size up to $M \times N/8$ bytes, and on average, only $M \times N/2$ LSBs in the image need to be changed. Images with more bits per pixel and multiple color channels have even larger capacity. Seemingly simple, LSB insertion in raw pixels is efficient and hard to detect visually. For compressed image formats (e.g., JPEG), LSB insertions is performed on the compressed data streams, for instance, the quantized DCT coefficients in a JPEG image. Similar to embedding in raw pixels, LSB insertion on the compressed data stream introduce negligible perceptual difference between the cover and stego images, Figure 6.2. Sophisticated steganographic softwares can add further layers of complexities, such as distributing messages in a pseudo-random way, avoiding embedding into low frequency smooth regions and encrypting messages. These measures make it even harder for an eavesdropper to detect the presence of the hidden message.

Steganography is closely related to cryptology. Both can be used for secure data communication. Also, in practice, steganographic messages can be encrypted before being embedded into the cover medium. However, the aim of steganography is to conceal the very existence of the

[1]Traditionally, an image with steganography embeddings is termed as a stego image.

Figure 6.2: Shown in the leftmost column is a color JPEG image. A message, which is a smaller image, was embedded into the cover using Jsteg (an implementation of LSB insertion in quantized DCT coefficients) to generate the stego images. Also shown in the rightmost column is absolute value of the difference between the cover and stego image, normalized into the range $[0, 255]$ for display purposes.

communicated message, while an encrypted message will certainly arouse suspicion. Digital watermarking [11] is another information hiding technique closely related to steganography. It aims to hide identification information in digital image for copyright protection or authorship authentication. Illegal duplication of the copyright protected image or tampering of the authentic image can be detected by checking the existence or lack of watermark. Steganography and digital watermarking are similar in the aim of hiding information. However, watermark for copyright protection are robust, so as to be detectable even in manipulated images. On the other hand, most steganography methods are fragile under image manipulations, therefore not appropriate for digital watermarking. Also, invisibility is not a primary design consideration of digital watermarking.

It worth noting that though hard to detect, there are easy ways to destroy potential steganographic messages, which can be achieved with simple image manipulations such as adding a small amount of noise without significantly altering the overall visual appearance. As steganography embeddings are fragile to image manipulations, such a "shaking bag" method [10] can effectively destroy the message hidden in an image. Nevertheless, there are situations when it is desirable to be able to detect the activity of steganography. For instance, knowing a suspect sending steganographic messages may be important to prevent further criminal activities.

Image Steganalysis

Steganography can be used as a covert communication method by criminals, terrorists and spies. Malicious message can be embedded into an innocuous-looking image, and posted on the Internet or sent in an email without being suspected. Therefore, it is not surprising that with the emergence of steganography, that the development of a counter-technology, steganalysis, has also emerged

(see [21] for a review). The goal of steganalysis is to determine if an image (or other carrier medium) contains an embedded message. As this field has developed, determining the length of the message [24] and the actual contents of the message are also becoming an active area of research.

Current steganalysis methods fall broadly into one of two categories: the embedding-specific approaches to steganalysis that take advantage of particular algorithmic details of the embedding algorithm [34, 83, 59, 22, 84], and generic steganalysis (e.g., [41, 23, 42]) that attempts to detect the presence of an embedded message independent of the embedding algorithm and, ideally, the image format. Embedding-specific methods are more efficient when the knowledge of embedding algorithm is present. For instance, LSB insertion in raw pixels results in specific changes in the image grayscale histogram, which can be used as the basis for its detection [84]. However, given the ever growing number of steganography tools[2], embedding-specific approaches are clearly not suitable in order to perform generic, large-scale steganalysis.

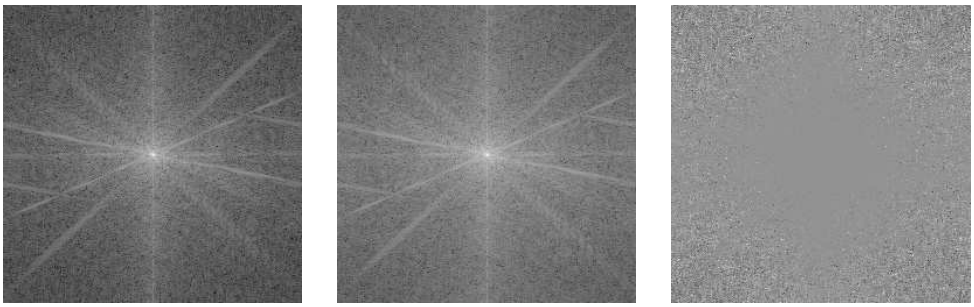
On the other hand, though visually hard to differentiate, the statistical regularities in the natural image as the steganography cover are disturbed by the embedded message. For instance, changing the LSBs of a grayscale image will introduce high frequency artifacts in the cover images. Shown in Figure 6.3 are (a) the logarithm of the Fourier transform of a natural image, and (b) the logarithm of the Fourier transform of the same image after a message is embedded into the LSBs. Shown in panel (c) is the difference between (a) and (b). All images are contrast enhanced for display. Note the difference between a clean and a stego image in the high frequency region, which are artifacts introduced by the embedding. The generic steganalysis (e.g., [41, 23, 42]) detects steganography by capturing such artifacts. We propose a general framework for generic image steganalysis, based on discriminative image features from the proposed image statistics and non-linear classification techniques. Without the knowledge of the embedding algorithm, our method detects steganography based on the abnormality in the statistics of the stego images. In the following, we describe our generic steganalysis systems for JPEG, TIFF and GIF images.

6.2 Generic Steganalysis of JPEG Images

JPEG Encoding and Decoding

Notwithstanding a lossy compression, JPEG [81] (short for Joint Photographic Experts Group) is the dominant image format currently for its high compression ratio. The overall process of JPEG encoding and decoding is shown in Figure 6.4 and 6.5, respectively. For encoding, the image is first divided into 8×8 blocks and shifted from unsigned integers to signed integers (to make the dynamic range of pixel values to be zero meaned). The blocks are then fed into the forward discrete cosine transform (FDCT). In decoding, the inverse DCT (IDCT) outputs 8×8 blocks to form the decompressed image. Denoting the image signal and its DCT decomposition as $I(x, y)$

[2] as listed at `www.jjtc.com/Steganography/toolmatrix`, there are over 200 different freely distributed steganography embedding programs, and more than 60% of them can be used to embed in images.



(a) (b) (c)

Figure 6.3: Shown are (a) the logarithm of the Fourier transform of a natural image, and (b) the logarithm of the Fourier transform of the same image after a message is embedded into the LSBs. Shown in panel (c) is the difference between (a) and (b). All images are contrast enhanced for display. Note the difference between a clean and a stego image in the high frequency region, which are artifacts introduced by the embedding.

and $F(u, v)$ respectively, the FDCT and IDCT are given in the following equations:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} I(x,y)\cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (6.1)$$

$$I(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v)\cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right), \quad (6.2)$$

where $C(u)$ takes value $1/\sqrt{2}$ for $u = 0$ and 1 otherwise. The DCT coefficients measure the relative amount of the 2D spatial frequencies contained in the 8×8 block. The 64 DCT coefficients are then subject to quantization with an 8×8 quantization matrix. Different compression quality will result in different quantization matrix, which stipulates the quantization step for each DCT coefficients. Quantization is the principle source of information loss in JPEG compression. After the quantization step, the DC coefficients ($F(0, 0)$ of each block is encoded with a linear prediction coder (LPC) – i.e., the DC coefficient of each block is replaced by the difference between adjacent blocks. Finally, all DCT coefficients are further encoded with a Huffman coder, and the data stream of a JPEG file contains the final encoded DCT coefficients.

JPEG Steganography

Most existing JPEG steganography tools are based on LSB insertions in the quantized DCT coefficients, as JPEG is a lossy compression algorithm. Embeddings in the intensity domain (e.g., LSB insertion in grayscales) will be destroyed by the lossy compression process, which discards high frequency components in image that are not essential for visual perception. On the other hand, the quantized DCT coefficient are subject to a lossless Huffman coding, which will not affect the embedded message. Besides, modifications to the LSBs of the quantized DCT coefficients introduce minimal visual artifacts to the cover image, Figure 6.2.

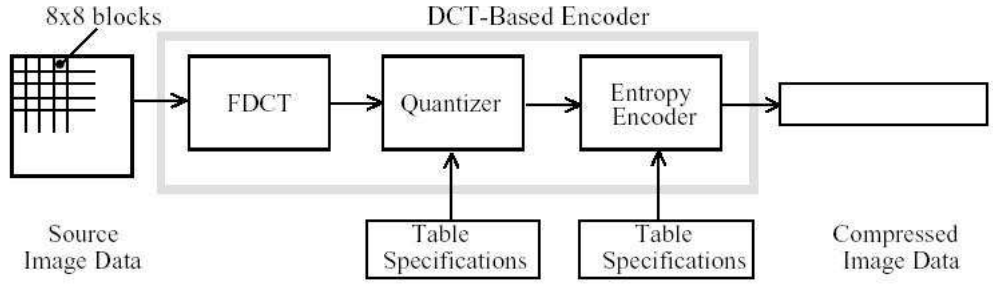


Figure 6.4: JPEG encoding process, fi gure from [81]

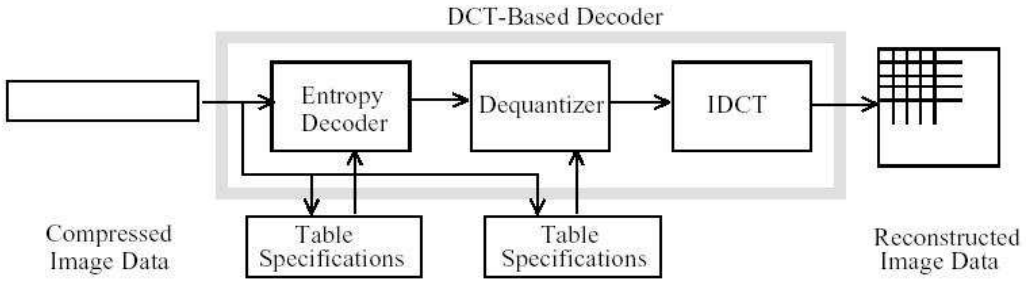


Figure 6.5: JPEG decoding process, figure from [81]

Based on this basic embedding scheme, different algorithms may implement further layers of complexities. For instance, most systems have the mechanism of distributing embedding locations based on a user provided stego key to make the pattern of embeddings irregular. Besides, to counter known steganalysis methods, many systems also manipulate the statistics of the stego image. For example, Outguess [60] only embeds into one-half of the redundant bits and use the remaining redundant bits to preserve the first-order statistics of the JPEG DCT coefficient distribution. F5 [22] uses a more sophisticated embedding algorithm which embeds message bits into randomly-chosen DCT coefficients. The embedding is not based on bit-replacement or exchanging any fixed pairs of values, but employs matrix embedding that minimizes the necessary number of changes to embed a message of certain length. It does not modify the histogram of DCT coefficient and keep some crucial characteristics of the histogram of a clean JPEG image file. All these made F5 harder to detect than previous embedding methods.

6.2.1 Experiments

The cover images in the experiments are the $40,000$ JPEG natural images described in Chapter 1. From these $40,000$ natural images, $40,000$ stego images ($1,600$ per embedding message type and per steganography tool) were generated by embedding random noise messages of various sizes into the full-resolution cover images. The messages were of sizes 6.0, 4.7, 1.2, 0.3 kilobytes (K),

corresponding to an average of 100%, 78%, 20% and 5% of the total steganography capacity[3] of the cover images, respectively. These messages were embedded using Jsteg [75], Outguess [60], Steghide [30], Jphide [38] and F5 [82]. When embedding a large message, its size might exceed the steganography capacity. In such a circumstance, a fraction of the message that fills the full capacity of the cover image was embedded. Each stego image was generated with the same quality factor as the original cover image so as to minimize double JPEG compression artifacts.

From each image, cover and stego alike, image feature vectors based on the proposed image statistics were collected. Specifically, six types of statistical features can be collected:

1. 72-D feature vector of grayscale local magnitude statistics;
2. 36-D feature vector of grayscale local phase statistics;
3. 108-D feature vector of grayscale local magnitude and local phase statistics;
4. 216-D feature vector of color local magnitude statistics;
5. 216-D feature vector of color local phase statistics;
6. 432-D feature vector of color local magnitude and local phase statistics.

To accommodate different image sizes, only the central 256×256 region of each image was analyzed.

We trained linear discriminant analysis (LDA), non-linear support vector machines with radial basis function (RBF) kernels, and one-class support vector machines (one-class SVM) with RBF kernel and multiple hyperspheres, based on the collected image statistics. The training sets for LDA and SVM consisted of image statistics from the $32,000$ randomly chosen natural images and $32,000$ randomly chosen stego images ($6,400$ per embedding program tested). The image statistics of the remaining cover and stego images were used to test the classifiers obtained – throughout, results from the testing stage are presented. In the training phase, the false-positive rate (i.e., the probability of a cover image being incorrectly classified as a stego image) was controlled to be less than 1%. This specific setting is the practical requirement of applying steganalysis, where the cost of a false positive is much higher than a false negative. The control over error rates were achieved by adjusting weights on different types of classification errors. The parameters of the RBF SVM, namely the width of the RBF kernel and the penalty factor, were tuned by cross-validation. The training set of the one-class SVM classifiers consisted of only the image statistics from the $32,000$ cover images, and none from the stego images. The image statistics of the remaining cover images and all stego images formed the testing set of the one-class SVM classifiers. For comparison, one-class SVMs with 1, 4, and 6 hyperspheres were trained. In each case, the false-positive rate was also fixed to be less than 1%.

[3]The steganography capacity is the rate between the size of embedded message and the maximum message size permitted of a cover image, which varies across different image formats, embedding softwares and cover images. The steganography capacity reported hereafter are based on the average maximum message size of 1000 cover images per embedding program.

In the following, we examine the general sensitivity and robustness of our approach to feature types, message size, the choice of classifier, the specific components of the statistical model, false-positive rate, JPEG compression, and to cover image format.

Feature Types

We first compare the performance of grayscale and color image statistics. Shown in Figure 6.6 are the detection accuracies for the LDA classifier, for five different steganography embedding programs and four different messages sizes and a training false positive rate of 1%, with (top) the 108 grayscale image statistics (72 local magnitude statistics and 36 local phase statistics) and (bottom) the 432 combined color image statistics (216 local magnitude statistics and 216 local phase statistics) for LDA classifiers on the same set of images. For point of comparison, the results for the color image statistics (bottom panel) are annotated with the detection accuracy for the grayscale image statistics (top panel). The left-most gray bar corresponds to the false-positive rate (a clean image incorrectly classified as stego), by subtracting it from 100%, which is, on average, less than 1.0%. This indicates that the LDA classifiers did not overfit the training data. For the grayscale image statistics, the average detection accuracy is 27.8%, 12.2%, 5.0% and 0.5% for embeddings at capacities 100%, 78%, 25% and 5%, with a maximum/minimum detection accuracy of $51.2\%/12.2\%$, $23.3\%/5.4\%$, $9.4\%/2.3\%$, and $1.0\%/0.1\%$. For the color image statistics, the average detection accuracy is 44.7%, 26.7%, 11.2% and 1.0% for embeddings at capacities 100%, 78%, 25% and 5%, with a maximum/minimum detection accuracy of $64.4\%/31.1\%$, $33.2\%/16.5\%$, $12.7\%/8.3\%$, and $2.3\%/0.3\%$. The color image statistics achieved, in general, higher accuracy than the grayscale images, as regularities across different color channels are considered. There is also a general trend that the classification accuracy decreases as the size of embedding message decreases, and the detection accuracy varies across different embedding programs - with F5 and Jsteg being the hardest and easiest, respectively.

Linear or Nonlinear Classification

The previous experiment confirms that the 432 color image statistics are more effective than the 108 grayscale image statistics for steganalysis. Next we compare different classification techniques combined with the color image statistics. Shown in Figure 6.7 (top) is the classification accuracy with the 432 color image statistics for five different steganography embedding programs and four different messages sizes and a training false positive rate of 1%, for non-linear SVM classifiers with RBF kernels, annotated with the detection accuracy of the LDA classifiers in Figure 6.6(bottom). The left-most gray bar corresponds to the false-positive rate (a clean image incorrectly classified as stego) by subtracting it from 100%, which is, on average, less than 1.0%, indicating the non-linear SVM classifiers did not overfit the training data. The average detection accuracy of the non-linear SVM classifier is 78.2%, 64.5%, 37.0% and 7.8% with a maximum/minimum detection accuracy of $91.1\%/66.4\%$, $76.4\%/51.8\%$, $43.2\%/31.3\%$ and $11.2\%/4.8\%$ for each embedding types. In general, the non-linear SVM classifier outperforms the linear classification with LDA, reflecting the flexibility of the non-linear SVM in learning a complex classification surface.

One-class SVM Classification

Though non-linear SVM with the color image statistics achieved the best performance, the data preparation and training process for non-linear SVMs are nevertheless complicated and vulnerable to those yet unknown embedding schemes. Specifically, the training set of a non-linear SVM must include all targeted steganography tools. On the other hand, the one-class SVM, section 3.3, has the advantage of requiring only the cover images in the training set. Shown in Figure 6.7 is the detection accuracy for a one-class SVM classifier with six hyperspheres (Section 3.3.2) for each of five steganography embedding programs, and four different messages sizes, with the 432 color image statistics and a training false positive rate of 1%. For point of comparison, the results for the one-class SVM (Figure 6.7(bottom)) are annotated with the detection accuracy for the linear SVM (Figure 6.7(top)). The one-class SVM has the average detection accuracy of 76.9%, 61.5%, 30.3% and 5.4%, with a maximum/minimum detection accuracy of $92.4\%/64.4\%$, $79.6\%/49.2\%$, $42.3\%/15.8\%$ and $8.9\%/2.7\%$. The one-class SVM results in only a modest degradation in detection accuracy, while affording a simpler training stage. The drawback, however, is that the training time for one-class SVM with multiple hyperspheres is much longer than the corresponding non-linear SVM classifier.

The use of the one-class SVM classifier with multiple hyperspheres bears the question of how many individual one-class SVM classifiers (or equivalently, hyperspheres) should be used. While multiple hyperspheres afforded a more compact support in the training stage, they tended to over-fit to the training data and led to poor generalization in the testing stage. We compared the performance of the one-class SVM classifiers with multiple hyperspheres. Shown in Figure 6.8 are the classification accuracy with color image statistics for one-class SVMs with different number of hyperspheres on embedding messages of 78% steganography capacity. The gray bars correspond to the false-positive rate (a clean image classified as stego), by subtracting them from 100%. Each group of four bars corresponds to different number of hyperspheres (hs) in the one-class SVM classifier. The horizontal axes correspond to steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)). Note that even though the overall detection improved with an increasing number of hyperspheres, the false-positive rates, 99.7,99.4 and 99.0, also increase considerably. The optimal number of hypersperes, thus, is a balance between the detection accuracy and the false-positive rate.

Categories of Image Statistics

Similar to the analysis in section 5.4, we would like to know the role of individual statistics and statistics category in the classification. Shown in Figure 6.9, from left to right, is the detection accuracy for a non-linear SVM trained with the 108 color coefficient marginal statistics only, the 108 magnitude linear prediction error statistics only, the 216 local phase statistics only, and the 216 local magnitude statistics including both coefficient marginal and magnitude linear prediction error statistics on embedding messages corresponding to 78% capacity. For point of comparison, the dots correspond to a non-linear SVM trained on the complete set of 432 color image statistics with both the local magnitude and local phase statistics. These results show that the combined local magnitude and local phase statistics provide for better detection accuracy than only a subset

of the statistics.

Next, we would like to investigate the contribution of the error statistics and marginal statistics in the 216 local magnitude statistics. Shown in Figure 6.10 is the accuracy of the classifier plotted against the number and category of statistics for the LDA classifier on the testing set composed of Jsteg stego images with embeddings of size 4.7 kilo-bytes with a 1% false positive rate[4]. Similar to section 5.3, we began by choosing the single image statistics, out of the 216 possible coefficient marginal and magnitude linear prediction error statistics, that achieved the best detection accuracy. This was done by building 216 LDA classifiers on each statistics, and choosing the one that yielded the highest accuracy (which was the variance in the error of the green channel's diagonal band at the second scale). We then chose the next best statistics from the remaining 215 statistics. This process was repeated until all statistics were selected. The solid line in Figure 6.10 is the detection accuracy as a function of the number of statistics added. The white and gray stripes correspond to magnitude linear prediction error and coefficient marginal statistics, respectively. If the statistics included on the i^{th} iteration was of coefficient marginal then a vertical gray line was drawn at the i^{th} position. Note that the coefficient marginal and magnitude linear prediction error statistics are interleaved, showing that both sets of statistics are important for classification. However, the latter part of the curve, where the classification accuracy plateaus correspond to the means and skewness, which are mostly close to zero and thus carry less classification information.

The specific contribution of the 216 local magnitude and 216 local phase statistics was analyzed similarly. Shown in Figure 6.11 is the detection accuracy of LDA classifiers as a function of the number and category of the 432 local magnitude and local phase statistics with a 1% training false positive rate. The white and gray stripes correspond to decomposition and local phase statistics, respectively. As the top 67 statistics are of the local magnitude, they have more influence in the classification. We were surprised that the phase statitics did not provide a larger boost to the overall detection accuracy. There are several possible reasons for this: (1) our specific statistical model for phase simply fails to capture the relevant phase statistics of natural images; (2) our phase statistics do capture the relevant phase statistics, but the steg embedding algorithms do not disturb these statistics; or (3) what we think most likely, the magnitude error statistics implicitly capture similar properties of the phase statistics – that is, geometric regularities (e.g., edges) are explicitly captured by the phase statistics through correlations between the angular harmonics, while these same regularities are implicitly captured by the error statistics through correlatins of the magnitude across space and scale.

False Positive Rates

An important factor in classification is the false positive rate, or the error rate for misclassifying a stego image as a clean image. Generally, the higher the false positive rate, the lower the corresponding classification accuracy. Shown in Figure 6.12 is the detection accuracy for a non-linear SVM with a 0.1% training false-positive rate. The average detection accuracy is 70.7%, 56.5%, 27.7% and 3.9% for embeddings at capacities 100%, 78%, 20% and 5%, with a maximum/minimum de-

[4]This analysis was performed only on the LDA because the computational cost of retraining $23,220 = 216 + \cdots + 1$ non-linear SVMs is prohibitive. We expect the same pattern of results for the non-linear SVM.

tection accuracy of $86.3\%/58.3\%$, $71.2\%/42.1\%$, $37.8\%/14.6\%$ and $7.1\%/1.3\%$. For point of comparison, these results are annotated with the detection accuracy for the non-linear SVM with a 1.0% false-positive rate, Figure 6.7(top). Note that an order of magnitude lower false-positive rate results in a relatively small degradation in detection accuracy.

JPEG Qualities

The JPEG quality is an important factor in the performance of steganalysis. Different JPEG quality corresponds to different quantization of the DCT coefficient, and has been shown in Chapter 4 to affect the classification based on the proposed image statistics. We tested the non-linear SVM classifiers, trained on cover and stego images of one JPEG quality, to cover and stego images of another JPEG quality. Specifically, we used the stego images with messages of steganography capacity 100% and 78%. Shown in Figure 6.13(top) is the detection accuracy for a non-linear SVM trained on JPEG images with quality factor 70 and then tested on JPEG images with quality 90. The dots in this panel is the classification accuracy of the same classifier on cover and stego images of JPEG quality 70. Shown in Figure 6.13(bottom) is the detection accuracy for a non-linear SVM trained on JPEG images with quality factor 90 and then tested on JPEG images with quality 70. The dots in this panel is the classification accuracy of the same classifier on cover and stego images of JPEG quality 90.

The classifier trained and tested on images with quality factor 90, achieved an average detection accuracy of 64.5% with a false-positive rate of 1.2%. When tested on images of quality factor 70, this same classifier achieved an average detection accuracy of 77.0%. This higher accuracy seems, at first glance, to be a bit puzzling, but note that the false-positive rate decreases to 77.4%, rendering this classifier largely useless for image qualities other than those near to the training images. The classifier trained and tested on images with quality factor 70 achieves an average detection accuracy of 54.4% with a false-positive rate of 1.2%. When tested on images of quality factor 90, this same classifier achieves an average detection accuracy of only 19.2%, with a false-positive rate of 8.7%, again rendering this classifier largely useless for image qualities other than those near to the training images. These results show that our classifiers do not generalize well to new JPEG quality factors, but that individually trained classifiers, on several JPEG quality factors, are able to detect steganography in cover images of varying compression factors.

Comparision with Previous Work

There are, of course, many steganalysis techniques that have emerged over the past few years. While many of these are specific to individual embedding programs, a few are universal, or near-universal approaches. In this section we attempt to compare the effectiveness of our approach to that of Fridrich [20], as it has clearly emerged has one of the most effective techniques.

Fridrich extracted statistical measurements based on marginal and joint DCT statistics, from clean and stego images. A Fisher linear discriminant classifier was then trained and tested on a collection of $1,800$ images. While there are some obvious high-level similarities to our approaches, a direct comparison is difficult since (1) Fridrich's approach was specifically designed to detect steganography in JPEG images while our approach was applied to JPEG, GIF, and TIFF formats;

(2) Fridrich employed a linear classifier while we employed linear and non-linear classifiers; (3) Fridrich tested her approach on $1,800$ grayscale images, while we tested ours on $40,000$ color images; and (4) Fridrich employed only 23 statistical features, while we employed a considerably larger 432 features.

With these caveats in mind, we compared the performance of our approaches on OutGuess and F5. For a 1% false-positive rate and an embedding rate for Outguess of 0.05 and 0.1 bpc [5] (bits per non-zero DCT coefficient), our detection accuracies (non-linear SVM) were 53.8% and 71.3% while those of Fridrich were 31.1% and 99.1%. For a 1% false-positive rate and an embedding rate for F5 of 0.05 and 0.1 bpc, our detection accuracies were 10.7% and 26.3% while those of Fridrich were 2.6% and 7.2%. While our approach seems to be more effective at lower embedding rates, Fridrich's approach is more effective at higher embedding rates. This is particularly impressive given the low-dimensional feature vector and the use of only a linear classifier.

6.3 Generic Steganalysis of TIFF and GIF Images

TIFF and GIF are two lossless image formats that can be used as covers for steganography. TIFF (Tagged Image File Format) is one of the most widely used lossless high-fidelity image formats. It is the *de facto* standard image format for high color depth (32-bit) imaging. As TIFF is an uncompressed image format, messages can be embedded with a generic LSB embedding that modulates the least-significant bits of a random subset of the pixel intensities. GIF (Graphic Interchange Format) was developed originally for images with 8-bit index colors, with a palette of 256 true colors. GIF is based on the lossless LZW compression. Steganography in GIF images can be implemented by LSB insertions of the color indexes (e.g., EzStego) or permutation of color palette entries (Gifshuffle).

To test the performance of the proposed image statistics and non-linear SVM classifier on TIFF and GIF images, we first converted from the $40,000$ JPEG natural image, the equal number of TIFF and GIF images[6]. For the TIFF cover images, messages were embedded using a generic least significant bit (LSB) algorithm. These messages were random noises of sizes $84.6K$, $75.0K$, $18.8K$ and $4.6K$ corresponding to embedding capacities of, approximately, 100%, 89%, 22% and 5%. For the GIF cover images, messages were embedded using EzStego [44] and Gifshuffle [37]. EzStego embeds message in GIF images by LSB insertion to the color index of each pixel, which slightly perturb the image. The messages embedded with EzStego were random noises of sizes $26.2K$, $22.7K$, $6.7K$ and $1.6K$, corresponding to embedding capacities of, approximately, 100%, 85%, 25% and 6%. Gifshuffle embeds message with a very different method. In Gifshuffle, each

[5] Our detection accuracies are given with respect to the total cover capacity, defined to be the maximum size of a message that can be embedded by the embedding algorithm. Comparable bpc values for these embedding rates were determined to allow for a direct comparison to Fridrich's results. For OutGuess, a bpc value of 0.05 and 0.1 corresponds to an embedding capacity of 44.2% and 88.5%, respectively. For F5, a bpc value of 0.05 and 0.1 corresponds to an embedding capacity of 7.8% and 15.7%, respectively.

[6] There is an argument that TIFF and GIF images converted from JPEG images make the subsequent steganalysis easier. However, in face of the relative scarce of high quality TIFF and GIF images available, this is still a viable solution and the results can predict those on raw TIFF or GIF images.

possible embedded message is represented by one specific order of the items in the color palette. For GIF images with 256 entries in the color palette, this means a total number of $256!$ of different messages, or a maximum $1.6K$ message can be embedded into a GIF image, regardless the size of the image. Perturbing only the color palette will not change the image signal, thus the proposed image statistics could not be applied directly. However, if the color palette for all GIF images were the same, which is the case in our experiment, we can treat the indexes as pixels of a grayscale image and employ grayscale image statistics for steganalysis. The messages embedded with Gifshuffle were random noises of sizes $1.6K$, $1.4K$, $0.4K$ and $0.08K$, corresponding to embedding capacities of, approximately, 100%, 85%, 25% and 6%.

Shown in Figure 6.14(top) is the detection accuracy for non-linear SVMs separately trained on the TIFF and GIF images. Each group of bars corresponds to a different embedding algorithm: from left to right, LSB (ls), EZStego (ez) and Gifshuffle (gs). The gray bars correspond to the false-positive rate of 1.0%. For the TIFF images, the detection accuracy is 72.3%, 52.9%, 11.3% and 1.2%. For the GIF images, the average detection accuracy is 62.9%, 47.8%, 17.2% and 1.7%. In terms of embedding capacity, these detection rates are slightly lower than the detection accuracy for JPEG cover images. Next, we applied the non-linear SVM classifiers trained on JPEG images to the TIFF and GIF images, for which the performance is shown in Figure 6.14(bottom). For the TIFF images, the detection accuracy is 72.3%, 52.9%, 11.3% and 1.2%. For the GIF images, the average detection accuracy is 62.9%, 47.8%, 17.2% and 1.7%. This shows that our classifiers do not generalize well to different image formats, but that individually trained classifiers, on different image formats, are able to detect steganography in cover images of different image formats.

6.4 Summary

In conclusion, our experiments confirm that the proposed image statistics and non-linear classification are effective in generic image steganalysis. Specifically, the 432 combined color image statistics (216 local magnitude statistics and 216 local phase statistics) and non-linear SVM seemed to have the best detection performance. On the other hand, a one-class SVM classifier with multiple hyperspheres had a slight degradation in detection accuracy yet a much simpler training process. We also observed that different false positive rates, JPEG qualities and image formats affected the detection accuracy. In building practical generic steganalysis systems, these factors must be considered. Finally, a similar generic steganalysis system can also be built for audio signals [32].

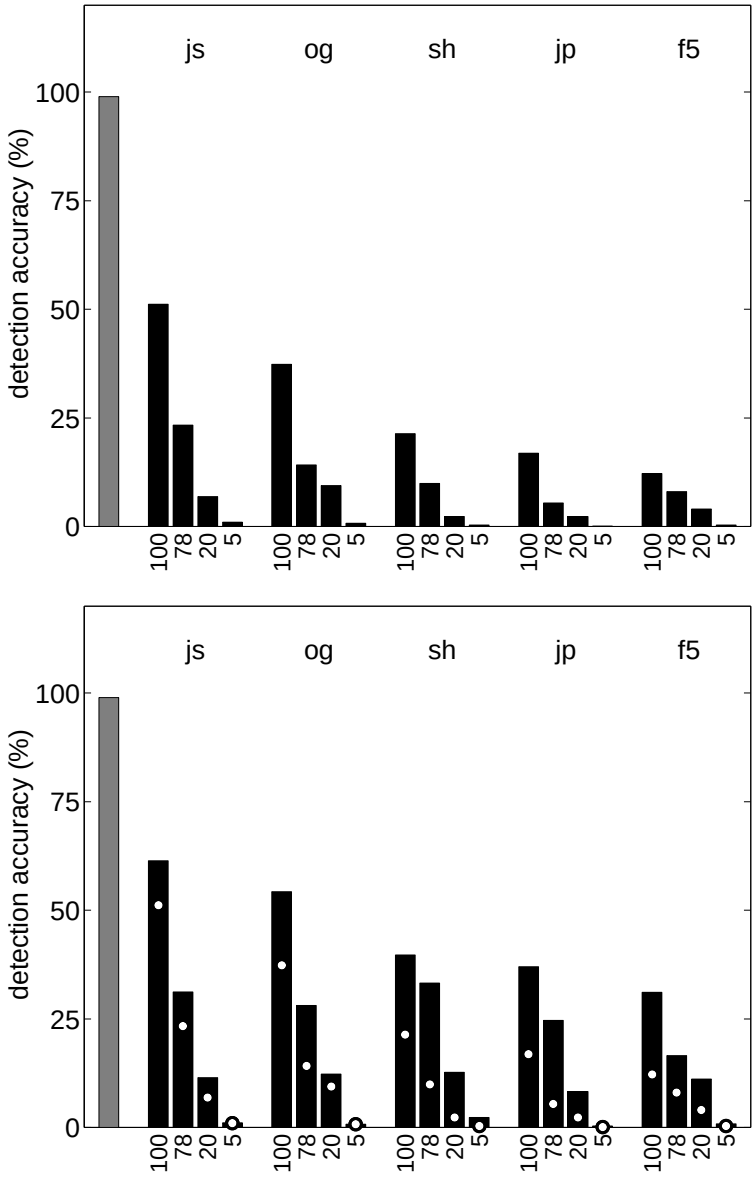


Figure 6.6: Classification accuracy for (top) 108 grayscale image statistics, and (bottom) 432 color image statistics, with LDA classification and a 1% false positive rate in training. The left-most gray bar corresponds to the false-positive rate (a clean image classified as stego), by subtracting it from 100%. Each group of four bars corresponds to different steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)). The numeric values on the horizontal axes correspond to the message size (as an average percentage of the steganography capacity of the covers). For point of comparison, the dots in bottom panel correspond to the detection accuracy of panel (top).

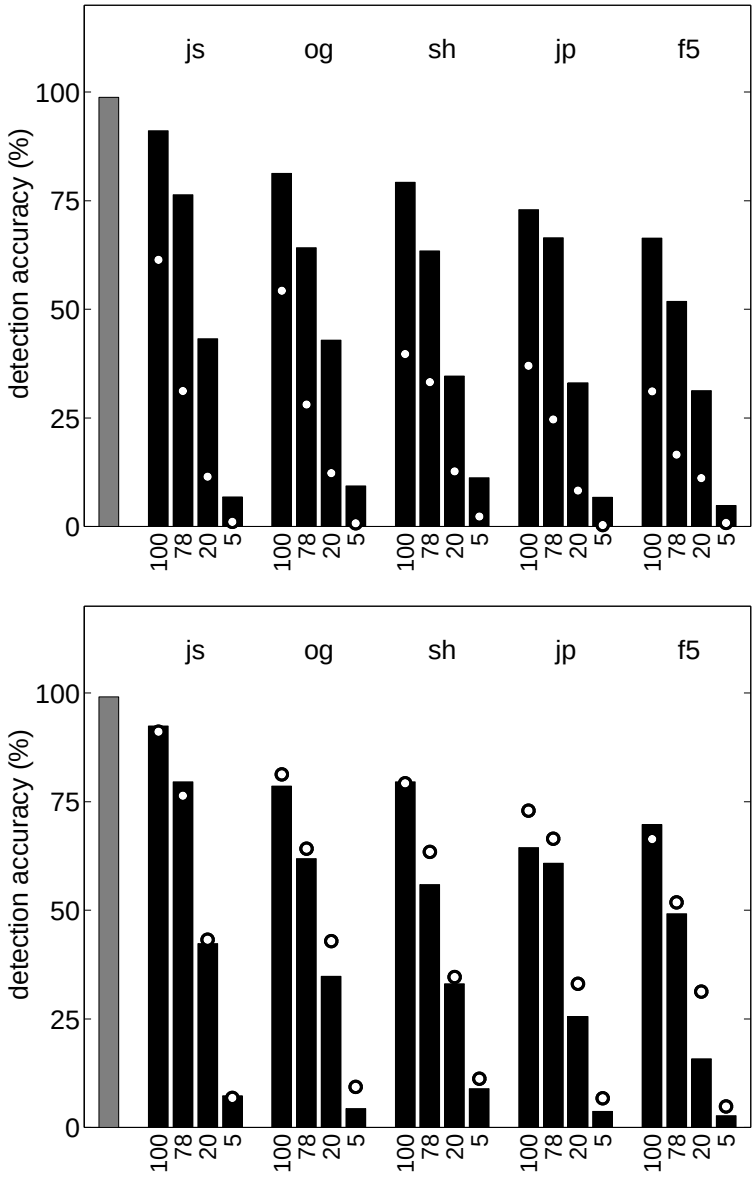


Figure 6.7: Classification accuracy with color image statistics for (top) non-linear and (bottom) one-class SVMs with a 1% training false positive rate. The left-most gray bar corresponds to the false-positive rate (a clean image classified as stego), by subtracting it from 100%. Each group of four bars corresponds to different steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)). The numeric values on the horizontal axes correspond to the message size as an average percentage of the steganography capacity of the covers. For point of comparison, the dots in bottom panel correspond to the detection accuracy in Figure 6.6 (bottom); and the dots in bottom panel correspond to the detection accuracy of top panel.

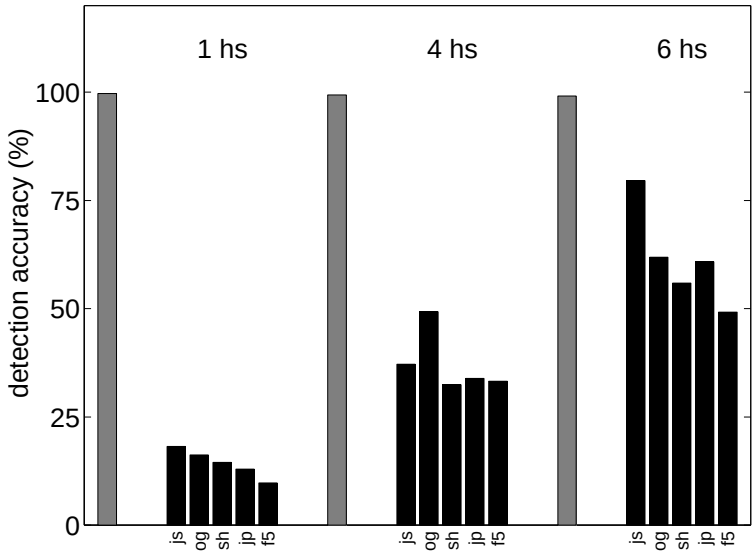


Figure 6.8: Classification accuracy with color image statistics for one-class SVMs with different number of hyperspheres on embedding messages of 78% steganography capacity. The gray bars correspond to the false-positive rate (a clean image classified as stego), by subtracting them from 100%. Each group of four bars corresponds to different number of hyperspheres (hs) in the one-class SVM classifier. The horizontal axes correspond to steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)).

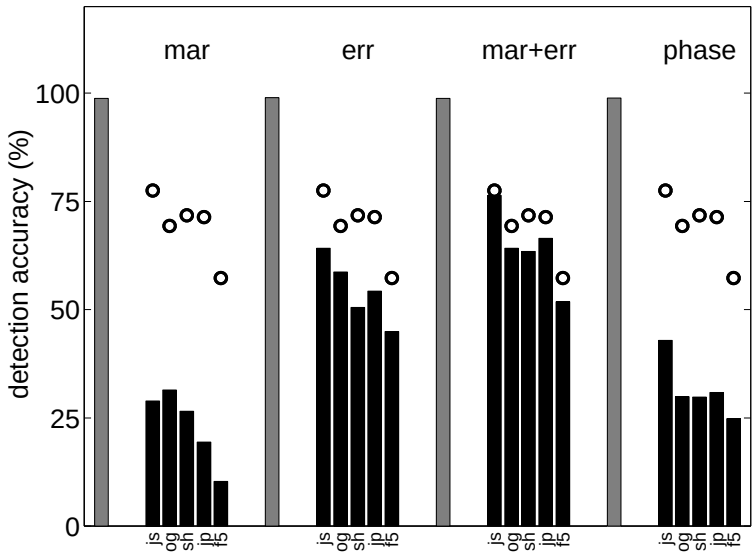


Figure 6.9: Classification accuracy for a non-linear SVM trained on (from left to right) the 108 color coefficient marginal statistics only, the 108 magnitude linear prediction error statistics only, the 216 local phase statistics only, and the 216 local magnitude statistics including both coefficient marginal and magnitude linear prediction error statistics on embedding messages corresponding to 78% steganography capacity. The dots correspond to a non-linear SVM trained on the complete set of 432 color image statistics with both the local magnitude and local phase statistics. The gray bar corresponds to the false-positive rate (a clean image classified as stego), by subtracting it from 100%. The horizontal axes correspond to different steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)).

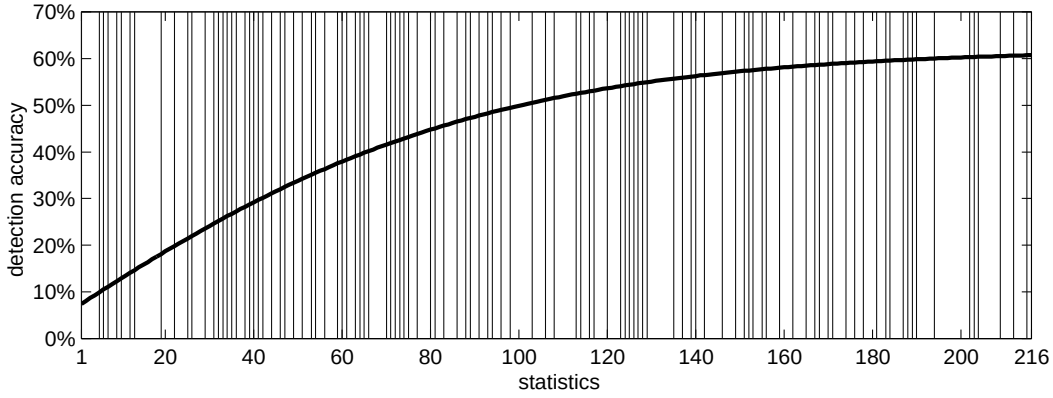


Figure 6.10: Shown is the detection accuracy of LDA classifiers as a function of the number and category of the 216 local magnitude statistics with a 1% training false positive rate. The horizental axis corresponds to the number of statistics incorporated, and the vertical axis corresponds to the detection accuracy in percentage. The white and gray stripes correspond to magnitude linear prediction error and coefficient marginal statistics, respectively.

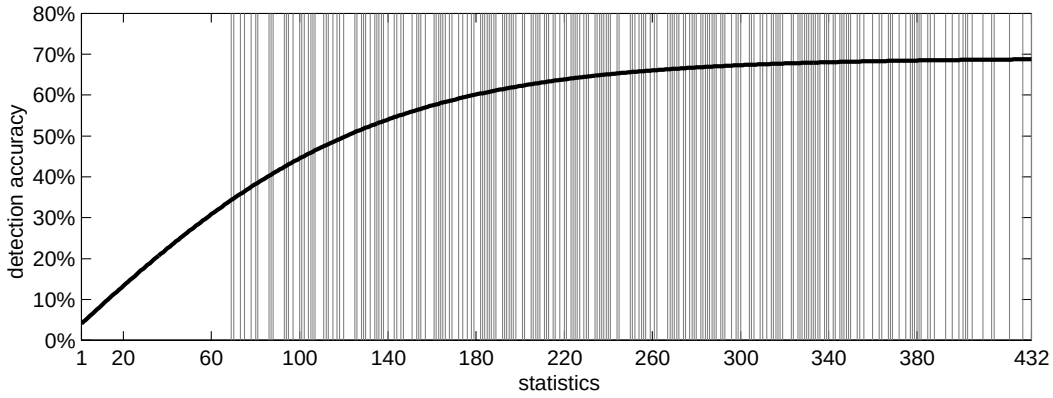


Figure 6.11: Shown is the detection accuracy of LDA classifiers as a function of the number and category of the 432 local magnitude and local phase statistics with a 1% training false positive rate. The horizontal axis corresponds to the number of statistics incorporated, and the vertical axis corresponds to the detection accuracy in percentage. The white and gray stripes correspond to decomposition and local phase statistics, respectively.

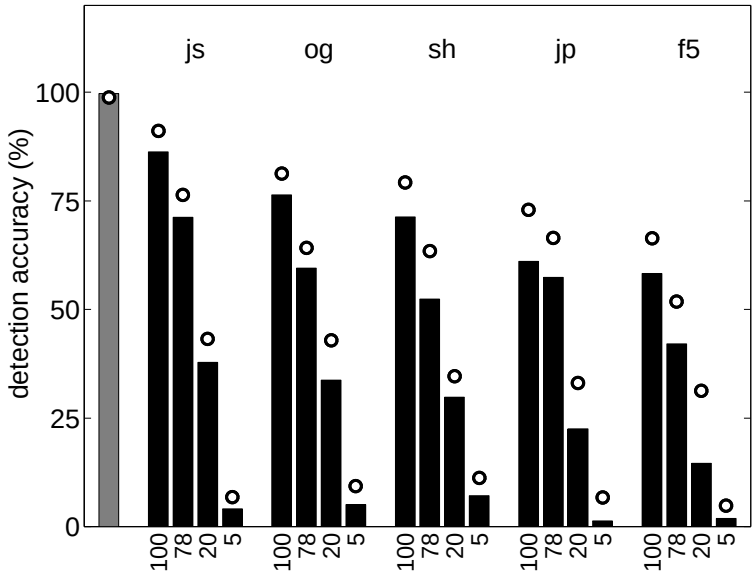


Figure 6.12: Classification accuracy for a non-linear SVM with color image statistics and with a 0.1% training false-positives. The dots correspond to the detection accuracy for a non-linear SVM with 1.0% false-positives, Figure 6.7(top). In all panels, the gray bar corresponds to the false-positive rate (a clean image classified as stego), by subtracting it from 100%. Each group of bars corresponds to different steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5)). The numeric values on the horizontal axes correspond to the message size as an average percentage of the steganography capacity of the covers.

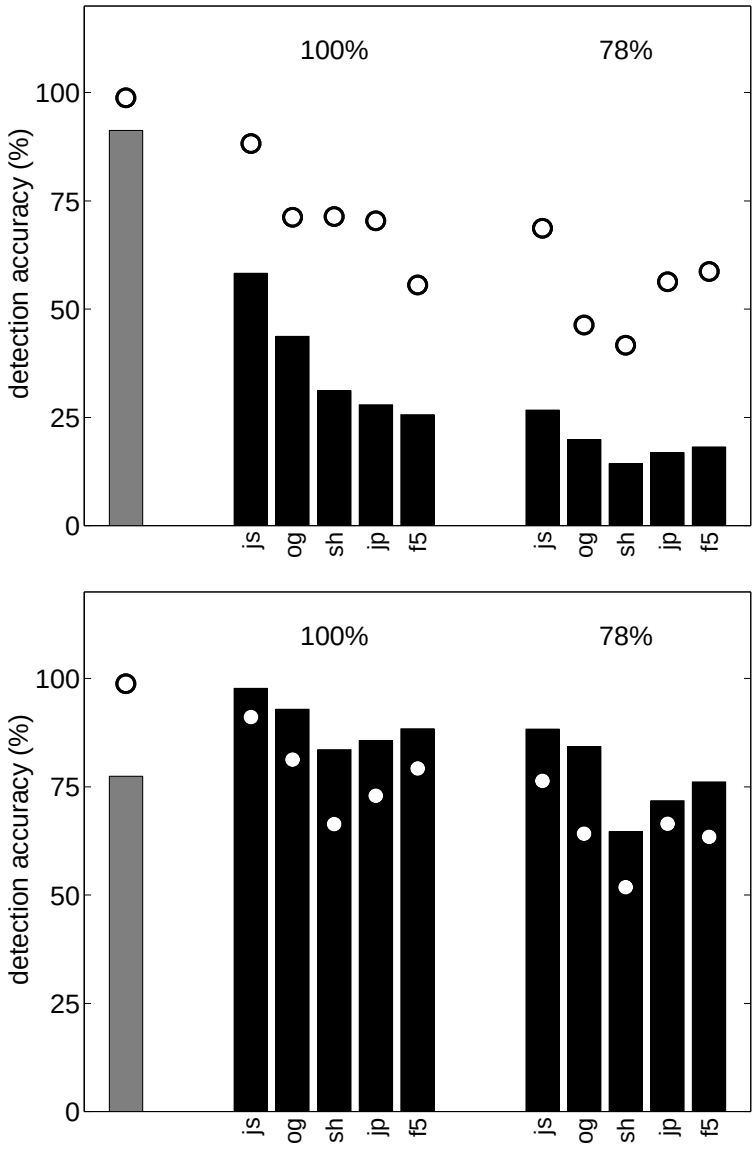


Figure 6.13: Classification accuracy for non-linear SVM classifiers (top) trained on JPEG images with quality factor 70 and tested on JPEG images with quality 90, and (bottom) trained on JPEG images with quality factor 90 and tested on JPEG images with quality 70. The dots in (top) correspond to the same classifier tested on quality factor 70, and the dots in (bottom) correspond to the same classifier tested on quality factor 90. The gray bars correspond to the false-positive rate (a clean image classified as stego), by subtracting it from 100%. Each group of bars corresponds to different embedding capacities, 100% and 78%. The horizontal axes correspond to steganography embedding programs (jsteg (js); outguess (og); steghide (sh); jphide (jp); and F5 (f5))

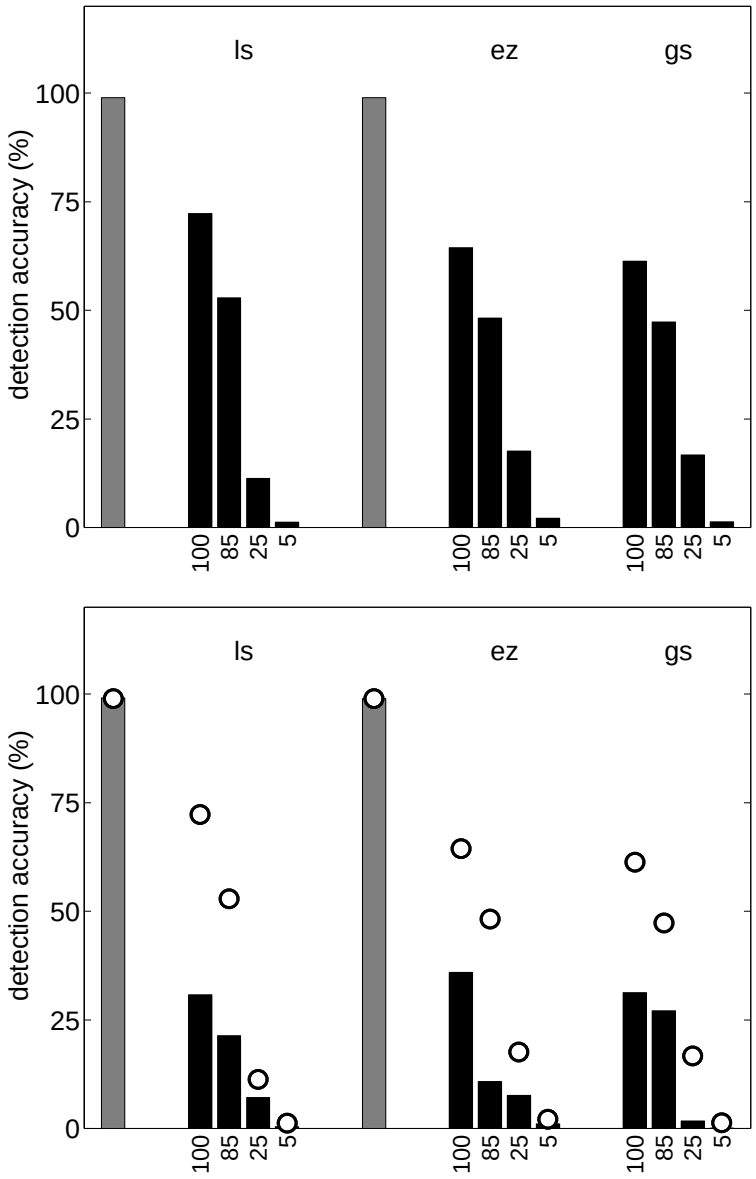


Figure 6.14: Classification accuracy on TIFF and GIF format images with a 1% training false-positive rate and for non-linear SVMs (top) with a training set consisted of TIFF or GIF images, and (bottom) a training set with JPEG images. The dots in (bottom) correspond to the accuracies in (top). The gray bars correspond to the false-positive rate (a clean image classified as stego), by subtracting it from 100%. Each group of four bars corresponds to different steganography embedding programs (a generic LSB embedding in TIFF (ls); EzStego in GIF (ez); and Gifshuffle in GIF (gs)). The numeric values on the horizontal axis correspond to the message size (as a percentage of cover capacity).

Chapter 7

Other Applications

We present in this chapter one another application of the proposed image statistics in digital image forensics, as for differentiating the rebroadcast (printed-out-and-scanned-in) images from the live captured images in section 7.1. In section 7.2, the proposed local magnitude image statistics (with a adapted neighborhood set) is applied to the traditional art authentication, where we differentiate or identify the artistic styles of different artists based on these image statistics.

7.1 Live or Rebroadcast

In recent years, biometric-based authentication (e.g., face, iris, voice or fingerprint) is increasingly gaining popularity in a large spectrum of applications, ranging from governmental programs (e.g., national ID card and visa) to commercial applications such as logical and physical access control. Compared to the traditional password-based authentication systems, the biometrics-based systems have the advantages of easier management (no need to memorizing the passwords and changing them periodically) and better security (biometrics confirm the identity of the user).

Surprisingly, however, even the most sophisticated biometric-based authentication systems may be vulnerable to a simple "rebroadcast" attack. For instance, to break an iris-recognition based authentication system, a malicious intruder can find a photograph of the face of an authenticated person, printed out on a paper (with sufficiently high printing quality), cut the iris images out and pasted on his eyelids. When being presented to the system, instead of the live captured iris image, Figure 7.1 top row, the system is fed with the rebroadcast iris images, Figure 7.1 bottom row. For the purpose of recognition, these images are just the same, with the rebroadcast images with different noise characteristics from the printing process. Since many biometric-based authentication systems are built to be robust in image noises, such a simple trick will, unfortunately, work for the intruder's purpose. It has been reported that two German hackers using a similar technique broke two commercial fingerprint authentication systems.

A key feature for biometric-based authentication systems withstanding the "rebroadcast" attacks, is to enable the systems to differentiate between a rebroadcast and a live image. Following the general methodology in Chapter 5 and 6, this problem is formulated as a binary classification where classifiers of live and rebroadcast images was built based on the proposed image statistics.

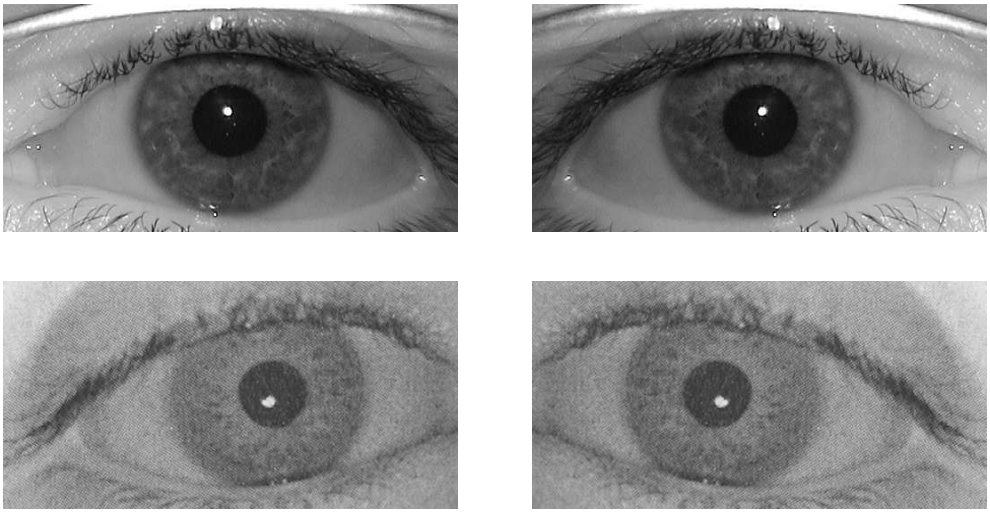


Figure 7.1: Shown is the original iris images (top row) and the images after being printed and scanned (bottom row). The mean of the absolute value of the difference between these images is 19 with a standard deviation of 17.5 (on a scale of $[0, 255]$).

Though previous chapters indicate that color image statistics and non-linear classification are the most effective solutions, we found that for this task, the simpler image feature comprised by the 72 grayscale local magnitude statistics and linear classifier (LDA) are sufficient for a good performance.

In our experiment, we randomly chose $1,000$ images from the $40,000$ color photographic images as described in Chapter 1. From these images, 200 "rebroadcast" images were generated by printing the grayscale converted photograph images on a laser printer and then scanned with a flatbed scanner, Figure 7.1. This is to simulate the condition when the printed-out images are captured by the camera in the authentication system. For consistency, printing and scanning were both done at 72 dpi (dots per inch) to reduce loss of image information as much as possible. Next, the 72 grayscale local magnitude image statistics were collected on all the $1,200$ images, from which 750 photograph images and 150 rebroadcast images were randomly chosen to form the training set. The remaining 250 photograph images and 50 rebroadcast images were included in the testing set.

In the training stage, the LDA classifier correctly classified 99.5% of the live and 100% of the rebroadcast images - with a threshold was selected so as to afford a less than 0.5% false positive rate. In the testing stage, 99.5% of the live and 99.8% of the rebroadcast images were correctly classiffed. To avoid reporting results pertaining to a specific training/testing split, these values were the average of results over 100 random training/testing splits. The relatively easy classification of live and rebroadcast images attributes to the artifacts introduced by the printing and scanning process, which the simple grayscale local magnitude statistics and linear classification suffice to capture. However, as the current experiment is only a rough simulation of the real rebroadcast attack, it may be the case when higher printing quality and camera resolution are used, more sophisticated image statistics (e.g., the 432 color image statistics) and non-linear classification

techniques (e.g., SVM) are needed. The trained classifier can then be used as a front-end to the biometric-based authentication system to pre-filter potential rebroadcast attacks.

7.2 Art Authentication

Many *chef d'œuves* in art are shrouded with mystery as the true identities of the artists, and art authentication aims to solve these mysteries. Of great interest to art authentication are the detection of forgeries, and the recovery of all artists involved in producing a piece (also known as the problem of "many hands"). Existing art authentication approaches largely rely on physical examination (e.g., X-rays or surface analysis) and/or the expertise of art historians. With the advent of powerful digital technology, it seems that computational tools can provide some new weapons into the arsenal of art authentication methods. Previous examples include the analysis of the fractal dimension of Jack Pollock's works [72].

The ability of the image statistics proposed in previous chapter to capture subtle difference between natural and unnatural images suggests that they may also be applied to differentiate the individual artistic styles in the scanned images of art works. These artistic styles, though fundamentally different from the image forensic applications, share the similarity that they are transcend simple difference in image contents, and intuitively justify the use of the proposed image statistics. Specifically, forgery detection is formulated as looking for difference in the image statistics between the works of different artists, and identification of involving artists as finding clustering in the resulting image statistics. In the following, we describe the application of the image statistics to the forgery detection for the works of Pieter Bruegel the Elder and the analysis of authorship in a painting of Perugino.

7.2.1 Bruegel

Pieter Bruegel the Elder (1525/30-1569) was perhaps one of the greatest Flemish artists. Of particular beauty are his landscape drawings. We choose to begin our analysis with Bruegel's work not only because of their exquisite charm and beauty, but also because Bruegel's work has recently been the subject of renewed study and interest [51]. As a result many drawings formerly attributed to Bruegel are now considered to belong to others. As such, we believe that this is a wonderful opportunity to test and push the limits of our computational techniques.

We digitally scanned (at 2400 dpi) two sets of drawings from the Bruegel folio from 35mm color slides, as listed in Table 7.1 (slides were provided courtesy of the Metropolitan Museum of Art [51]): the portfolio consisted of eight authenticated landscape drawings by Bruegel and five forgeries. Shown in Figure 7.2 are images of an authentic Bruegel drawing (catalog #6) and a forgery (catalog #7). These color (RGB) images, originally of size 3894×2592, were cropped to a central 2048×2048 pixel region, converted to grayscale using Gray = 0.299Red + 0.587Green + 0.114Blue[1], and auto-scaled to fill the full intensity range $[0, 255]$.

[1] Although converting from color to grayscale results in a signifi cant loss of information,we did so to make it more likely that the measured statistical features and subsequent classification were more likely to be based on the pen

Figure 7.2: Authentic #6 (top) and forgery #7 (bottom), see Table 7.1.

For each of 64 (8×8) non-overlapping 256×256 pixel region in each image, the proposed image statistics were collected. Empirically, we found that the 72 grayscale local magnitude statistics with adapted neighborhood sets (section 2.2.1) achieved sufficiently good performance for authenticating these Bruegel paintings. Each image of a painting in question was thus reduced to a set of 64 72-D image feature vectors. In order to determine if there is a statistical difference between the authentic drawings and the forgeries, we computed the Hausdorff distance (Appendix A) between all pairs of images. The Hausdorff distance is defined on two sets of high-dimensional vectors and is shown to be robust to outliers in the sets. The resulting distance matrix was then subjected to a metric multidimensional scaling (MDS) (Appendix B), which help to visualize the data in a lower-dimensional space while best preserving their pairwise Hausdorff distances. Shown in Figure 7.3 is the MDS result of visualizing the 13 images, with the Euclidean distances between each pair of points corresponding to the Hausdorff distances. The circles correspond to the authentic drawings, and the squares to the forgeries. For purely visualization purposes, the wire-frame sphere is rendered at the center of mass of the authentic drawings and with a radius set to fully encompass all data points corresponding to the authentic paintings. Note that in both cases, the forgeries fall well outside of the sphere. For the first set of 8 authentic and 5 forged drawings, the distances of the authentic drawings to the center of the sphere are 0.34, 0.35, 0.55, 0.90, 0.56, 0.17, 0.54, and 0.85; and the distances of the forgeries are considerably larger at 1.58, 2.20, 1.90, 1.48, and 1.33. The means of these two distance populations are statistically significant, both having p-values with $p < 1^{-5}$ (one-way ANOVA). These results suggested that even in this reduced dimensional space, there is a clear difference between the authentic drawings and the forgeries.

We next investigate, with a subset of the authentic and forged Bruegel paintings, the roles played by the coefficient marginal and magnitude linear prediction error statistics in differentiating the artistic styles. Figure 7.4 shows the results of applying MDS with the coefficient marginal (left) and magnitude linear prediction error statistics. That is, for each of the sets of 36-D vectors of coefficient marginal statistics and magnitude linear prediction error statistics we build separate distance matrices (using Hausdorff distance) and apply MDS for a three-dimensional visualization. Notice that after this process the marginal statistics no longer appear to separate the drawings

strokes and not on simple color differences.

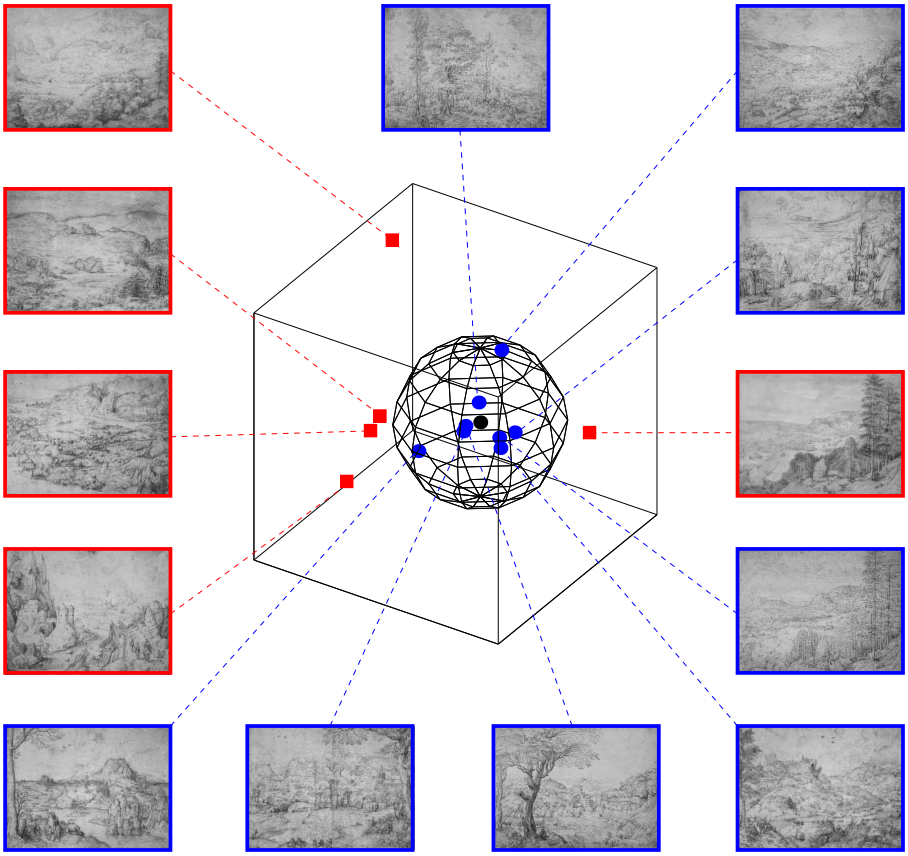


Figure 7.3: Results of the MDS analysis of the eight authentic Bruegel drawings (dots) and five forgeries (squares) of landscapes. The wire-frame and the central black dots are added for better visualization. Note in both cases, there is a clear difference between the authentic drawings and the forgeries.

while the error statistics do still succeed in achieving good separation, justifying their importance in differentiating the underlying artistic styles.

7.2.2 Perugino

A more challenging problem than forgery detection is to find out how many different artists participated in producing an art work. Many art works were the results of many artists collaboration, but with only the most prominent among them being accredited. It is, therefore, only fair to shed light on those un-named artists and gave them respect they deserved.

Pietro di Cristoforo Vannucci (Perugino) (1446-1523) is well known as a portraitist and a fresco painter, but perhaps best known for his altarpieces. By the 1490s, Perugino maintained a workshop in Florence as well as in Perugia and was quite prolific. Shown in Figure 7.5 on the left is the painting *Madonna with Child* by Perugino. As with many of the great Renaissance paintings, however, it is likely that Perugino only painted a portion of this work and his apprentices (among

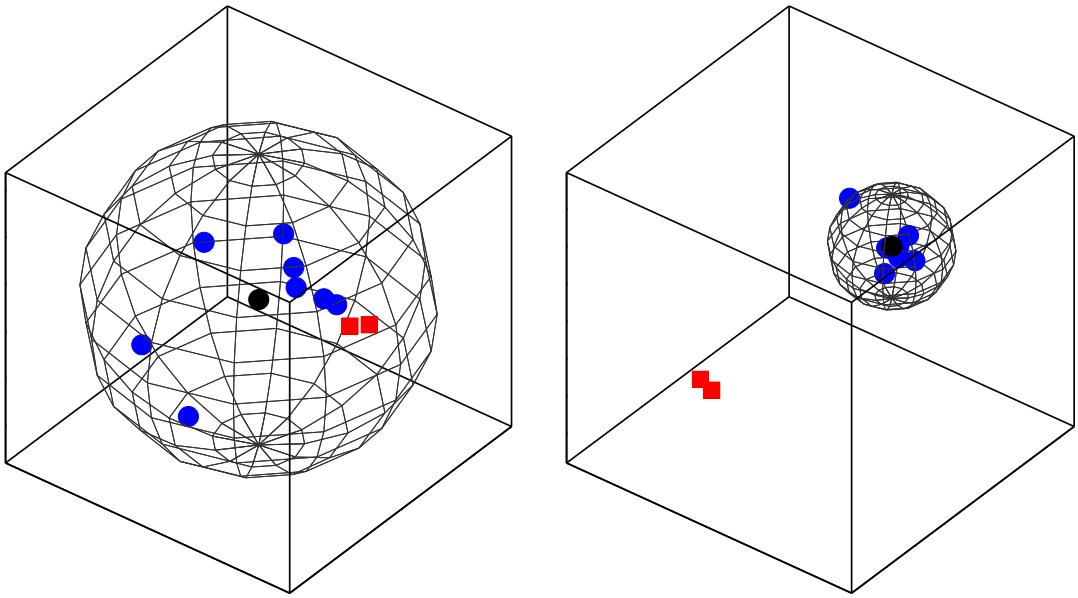


Figure 7.4: Application of MDS separately to the second set of eight authentic Bruegel drawings (dots) and two forgeries (squares) with the coefficient marginal statistics (left) and the magnitude linear prediction error statistics (right). The wire-frame and the central black dots are added for better visualization. The error statistics still achieve separation while the marginal statistics do not.

whom was the later famous Raffaello Sanzio) did the rest. To this end, we wondered if we could uncover statistical differences among the faces of the individual characters to recover how many different artists contributed to this work.

The painting (courtesy of the Hood Museum, Dartmouth College), top in Figure 7.5,was photographed using a large-format camera (8×10 inch negative) and drum-scanned to yield a color 16852×18204 pixel image. As in the previous section this image was converted to grayscale. The facial region of each of the six characters was manually localized and cropped, Figure 7.5 bottom left. Each face was then partitioned into non-overlapping 256×256 regions and auto-scaled into the full intensity range $[0, 255]$. This partitioning yielded 189, 171, 189, 54, 81, and 144 regions for each face numbered as in Figure 7.5 bottom left. Similar to the case of the the analysis of the Bruegel's works, the 72 grayscale local magnitude statistics with adapted neighborhood sets were collected from each of these regions. Then, we computed the Hausdorff distance between all six faces. The resulting 6×6 distance matrix was then subjected to MDS. Shown in Figure 7.5, bottom right, is the result of visualizing the original six faces with the top-three MDS eigenvalue eigenvectors. The numbered data points correspond to the six faces in the bottom left panel of Figure 7.5. Note how the three left-most faces cluster, while the remaining faces are distinct. The average distance between these faces is 0.61, while the average distance between the other faces is 1.79. This clustering pattern suggests the presence of four distinct hands, and is consistent with the views of some art historians [1].

Appendix A: Hausdorff Distance

The Hausdorff distance is a distance metric defined on two sets of vectors, X and Y, as

$$\mathrm{H}(X,Y) = \max\{h(X,Y), h(Y,X)\}, \tag{7.1}$$

where h is defined as

$$h(X,Y) = \max_{\vec{x}\in X}\{\min_{\vec{y}\in Y} d(\vec{x},\vec{y})\}. \tag{7.2}$$

Here $d(\cdot,\cdot)$ can be any distance metric defined on the vector space subsuming X and Y and in our case, we used the Euclidean distance in the 72-D vector space.

Appendix B: Multidimensional Scaling

Multidimensional scaling (MDS) is a popular method to visualize high dimensional data. Given N vectors $\{\vec{x}_1,\ldots,\vec{x}_N\}$, where $\vec{x}_i \in \mathcal{R}^d$, the goal of MDS is to find a lower-dimensional embedding for these data that minimally distorts their pairwise distances. Denote the $n \times n$ distance matrix as $D_{it} = d(\vec{x}_i,\vec{x}_j)$, where $d(\cdot,\cdot)$ is a distance metric in $\mathcal{R}^d$. The most common such metric is Euclidean distance defined as $d(\vec{x}_i,\vec{x}_j) = (\vec{x}_i-\vec{x}_j)^T(\vec{x}_i-\vec{x}_j)$.

Given the pairwise symmetric distance matrix, the classical (metric) MDS algorithm is given by the following steps:

1. Let $A_{ij} = -\frac{1}{2}D_{ij}^2$.
2. Let $B = HAH$, where $H = I_N - \frac{1}{N}\vec{u}\vec{u}^T$, I_N is the $N \times N$ identity matrix, and each component of the N-dimensional vector $\vec{u}$ is 1.
3. Compute the eigenvectors, $\vec{e}_1,\ldots,\vec{e}_N$ and corresponding eigenvalues $\lambda_1,\ldots,\lambda_N$ of matrix B, where $\lambda_1 \geq \cdots \geq \lambda_N$.
4. Taking the first d' eigenvectors to form matrix $E = [\vec{e}_1,,\ldots,\vec{e}_{d'}]$. The row vectors of E provide the d'-dimensional representations that minimally distort the pairwise distance of the original higher dimensional data.

Num.	Title	Artist
3	Pastoral Landscape	Bruegel
4	Mountain Landscape with Ridge and Valley	Bruegel
5	Path through a Village	Bruegel
6	Mule Caravan on Hillside	Bruegel
9	Mountain Landscape with Ridge and Travelers	Bruegel
11	Landscape with Saint Jermove	Bruegel
13	Italian Landscape	Bruegel
20	Rest on the Flight into Egypt	Bruegel
7	Mule Caravan on Hillside	-
120	Mountain Landscape with a River, Village, and Castle	-
121	Alpine Landscape	-
125	Solicitudo Rustica	-
127	Rocky Landscape with Castle and a River	Savery

Table 7.1: List off authentic (top) and forgeries (bottom) of Bruegel paintings. The first column corresponds to the catalog number in [51].

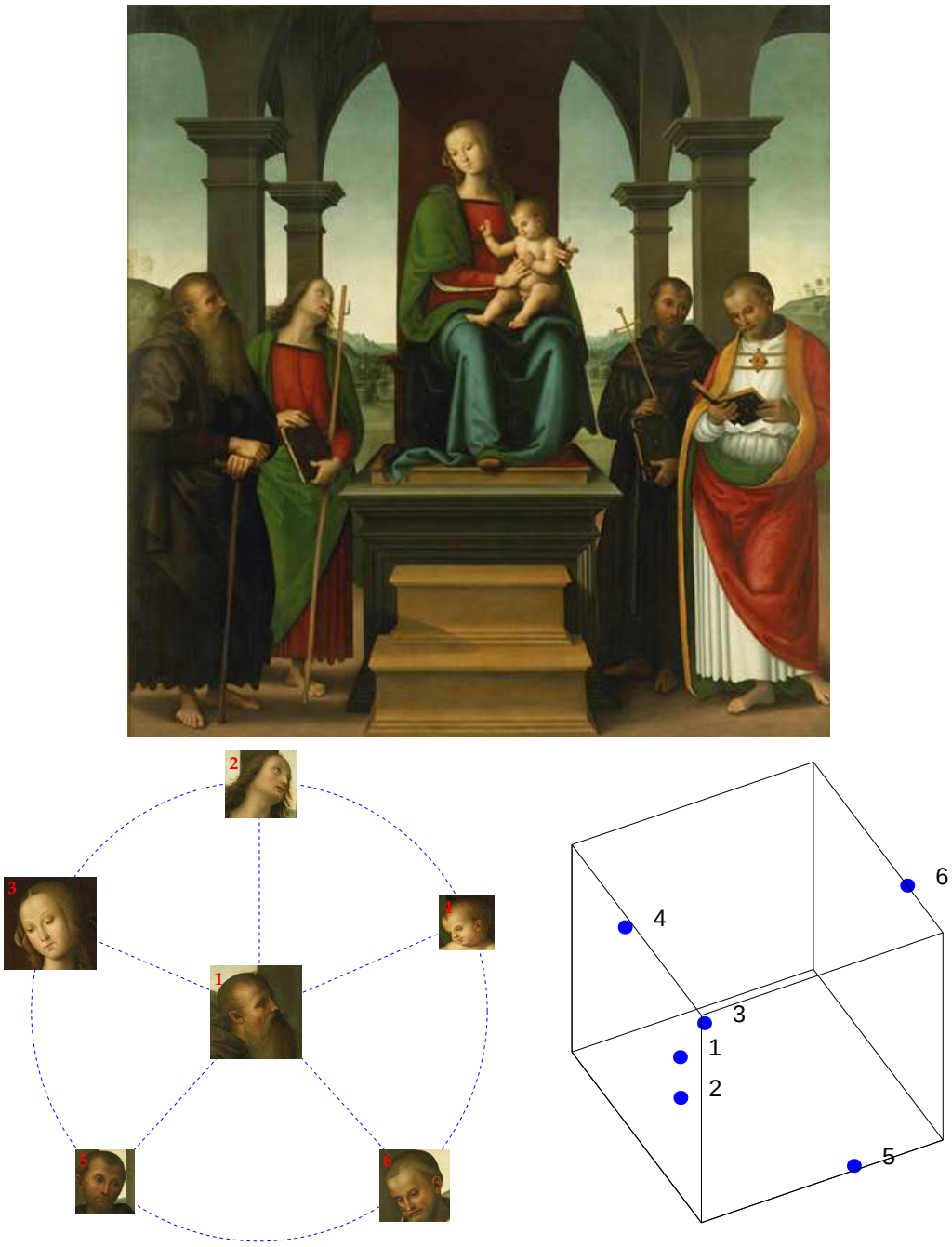


Figure 7.5: Left: *Madonna with Child* by Perugino (top). How many hands contributed to this painting? Bottom left: faces in the painting being analyzed. Bottom right: results of analyzing the Perugino painting. The numbered data points correspond to the six faces in the bottom left panel. Note how the first three faces cluster, while the remaining faces are distinct. This clustering pattern suggests the presence of four distinct hands.

Chapter 8

Discussion

Natural images are ubiquitous, yet they are relatively rare in the sea of all possible images. More importantly, natural images have statistical regularities that has been employed by the biological vision systems. Capturing such statistical regularities of natural images are essential for many applications in image processing, computer vision and especially, digital image forensics. We have presented, in this thesis, a set of image statistics based on the multi-scale image decompositions, which correspond to image representations with basis functions localized in both spatial and frequency domains. Such image representations are shown to be suitable to reveal statistical regularities in natural images. Specifically, our image statistics consist of the local magnitude statistics from a quadrature mirror filters (QMF) pyramid decomposition and the local phase statistics from a local angular harmonic decomposition. Empirically, we showed that these image statistics can differentiate natural images from some simple "un-natural" synthetic images. We then demonstrated their effectiveness, combined with non-linear classification techniques, in applications in digital image forensics on: (1) differentiating photographic and computer-generated photorealistic images (photographic vs. photorealistic); (2) detecting hidden messages in photographic images (generic steganalysis); and (3) identifying printed copies of photographic images to protect the biometric-based authentication systems from the rebroadcast attack (live vs. rebroadcast). All these techniques follow a unified framework of image classification: first feature vectors based on the proposed image statistics are collected on a set of labeled images, then a classifier is trained on these feature vectors and used to determine the class of an unknown image. The proposed image statistics are also applied to the traditional art authentication, where they were employed to differentiate authentic and forged drawings of Peter Bruegel the Elder and determine the artists involved in creating Perugino's "Madonna with Child".

Though effective in these practical applications, the work presented in this thesis has still a lot of room for further improvement.

Image Statistics First, the proposed image statistics are based on empirical observations of natural images and not derived directly from first principles of physical imaging process or neural processing in biological vision systems. Therefore, the image classification based on these image statistics cannot be fully and rigorously explained yet. It is therefore desirable to set this work on a firmer theoretical foundation, where

the proposed image statistics are approximations of more complicated models of natural images. This will not only advance our understanding of these image statistics, but also make it possible to extend them for wider applications.

Also, from the counter point of view of digital forensics, it will be interesting to know if it is possible to invert these image statistics, so that an unnatural image (e.g., a photorealistic image, or an image with steganography embedding, or a rebroadcast image) will not be detect in subsequent classification. This requires a way to manipulate image with minimum perturbation of image contents, but will result in similar image statistics as a natural image. Such an inverting procedure will shed light on the role of the proposed image statistics in characterizing natural images, and will stimulate more sophisticated tools in digital image forensics. Right at the heart of such an inverting procedure will be a sampling procedure similar to the one in [57], where natural images can be generated by sampling with constraints on their image statistics. Although it is not hard to generate images consistent with coefficient marginal statistics in a multi-scale image decomposition, constraints on the linear prediction errors and local phases make constructing such an inverting procedure hard.

An even harder but more significant improvement to the current work is to build generative statistical models based on the proposed statistics. A generative statistical model is the holy grail of natural image statistics research, from which images can be sampled and generated. Having such a generative model is to have a holistic picture of all natural images, thus it is utmost important. While the primary use of the proposed images is as discriminative image features in image classification, insights on generative models may be obtained by building generative model from the proposed image statistics.

In solving the digital image forensics problems in this thesis, we exclusively rely on the proposed image statistics. However, we are well aware of other statistical image features and the possibility of including them into the system for better performance. In this work, we focus on using image statistics that are generic and able to capture statistical regularities of all natural images. It is for this reason that we intentionally avoid using image statistics that are potentially useful for specific applications. When building a realistic system, these considerations are not so relevant and the proposed image statistics are expected to achieve better performance with the incorporation of other more domain-specific features.

Photographic vs. Photorealistic Our experimental results show that the proposed image statistics can be used to differentiate photographic and computer-generated photorealistic images. Currently, this only work for classification a whole image as either photographic or photorealistic. There are images that are combination of photographic and photorealistic images, as movie makers employ computers to generate part of the scenes. It is therefore interesting to extend the classification to the local level, where the parts generated by computer in an image can be detected. This is useful in digital image forensics to detect tampering of a photographic image with computer generated

regions. Another possible improvement in this direction is to construct a subjective measure of photorealism in digital images based on the proposed image statistics.

Generic Steganalysis From the proposed image statistics and non-linear classification, generic steganalysis can be built for JPEG, TIFF and GIF images. However, a true generic steganalysis system will also work across different image formats. The proposed image statistics have components that are sensitive to changes in image formats, thus not suitable for such a purpose. Therefore, an important improvement to this work is to find image statistics that are relatively stable in face of different image formats, and yet are suitable to the purpose of steganalysis.

Other Applications Besides the digital image forensics and art authentication, the proposed image statistics, with their ability to capture regularities in natural photographic images, have also other applications in image processing and analysis. One such application is automatic determination of the orientation (landscape or portrait) of an image [40], which is based on the fact that the local magnitude statistics for vertical and horizontal subbands will switch position for a 90° rotated image. Potentially, these image statistics can also be applied to image retrieval, general image classification and scene categorization.

In conclusion, we believe that statistical characterization of natural images is at the heart of many challenging problems in image processing and analysis, computer vision, and digital image forensics. The work described in this thesis reveals just a small fraction of the tremendous potential of these image statistics. Though still having a lot of room for improvement, they can serve as a well-posed starting point for future exploration in this direction.

Bibliography

[1] Personal correspondence with Timothy B. Thurber, Hood Museum, Dartmouth College.

[2] R. J. Anderson and F. A. P. Petitcolas. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, 16(4):474–481, 1998.

[3] V. Athitsos, M. J. Swain, , and C. Frankel. Distinguishing photographs and graphics on the world wide web. In *Workshop on Content-Based Access of Image and Video Libraries (CBAIVL)*, Puerto Rico, 1997.

[4] J. D. Bonet and P. Viola. A non-parametric multi-scale statistical model for natural images. In *Advances in Neural Information Processing Systems*, 1997.

[5] J. D. Bonet and P. Viola. A multi-scale multi-orientation feature for content-based image retrieval. In *IEEE Conference Computer Vision and Pattern Recognition*, 2000.

[6] R. W. Buccigrossi and E. P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8(12):1688–1701, 1999.

[7] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[8] G. Carneiro and A. Jepson. Phase-based local features. In *European Conference on Computer Vision (ECCV)*, 2001.

[9] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[10] E. Cole. *Hiding in Plain Set: Steganography and the Art of Covert Communication*. Wiley, 2003.

[11] I. Cox, M. Miller, and J. Bloom. *Digital Watermarking*. The Morgan Kaufmann Series in Multimedia and Information Systems. Morgan Kaufmann, 2001.

[12] G. Cross and A. Jain. Markov random field texture models. *IEEE Transactions on Pattern and Machine Intelligence*, 5:25–39, 1983.

[13] F. Cutzu, R. Hammoud, and A. Leykin. Estimating the degree of photorealism of images: Distinguishing paintings from photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[14] J. G. Daugman. Entropy reduction and decorrelation in visual coding by oriented neural receptive fields. *IEEE Transaction on Biomedical Engineering*, 36(1):107–114, 1989.

[15] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[16] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.

[17] R. Fletcher. *Practical Methods of Optimization*. John Wiley and Sons, 2nd edition, 1987.

[18] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1997.

[19] W. T. Freeman and E. C. Pasztor. Learning low-level vision. In *IEEE International Conference on Computer Vision*, Corfu, Greece, 1999.

[20] J. Fridrich. Feature-based steganalysis for jpeg images and its implications for future design of steganographic schemes. In *Proc. 6th Information Hiding Workshop,* Toronto, Canada, 2004.

[21] J. Fridrich and M. Goljan. Practical steganalysis: State of the art. In *SPIE Photonics West, Electronic Imaging*, San Jose, CA, 2002.

[22] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[23] J. Fridrich, M. Goljan, and D. Hogea. New methodology for breaking steganographic techniques for JPEGs. In *SPIE Symposium on Electronic Imaging*, Santa Clara, CA, 2003.

[24] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal. Quantitative steganalysis of digital images: Estimating the secret message length. *ACM Multimedia Systems Journal, Special issue on Multimedia Security*, 9(3):288–302, 2003.

[25] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern and Machine Intelligence*, 6:721–741, 1984.

[26] J. Gluckman. On the use of marginal statistics of subband images. In *IEEE International Conference on Computer Vision*, Nice, France, 2003.

[27] E. Hadjidemetriou, M. Grossberg, and S. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, 2004.

[28] M. Hassner and J. Sklansky. The use of Markov random fields as models of texture. *Computer Graphics and Image Processing*, 12:357–370, 1980.

[29] M. Heiler and C. Schnörr. Natural image statistics for natural image segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, Nice, France, 2003.

[30] S. Hetzl. *Steghide*. `steghide.sourceforge.net`.

[31] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[32] M. K. Johnson, S. Lyu, and H. Farid. Steganalysis in recorded speech. In *SPIE Symposium on Electronic Imaging*, San Jose, CA, 2005.

[33] N. Johnson and S. Jajodia. Exploring steganography: seeing the unseen. *IEEE Computer*, 31(2):26–34, 1998.

[34] N. Johnson and S. Jajodia. Steganalysis of images created using current steganography software. *Lecture notes in Computer Science*, 1525:273–289, 1998.

[35] D. Kahn. The history of steganography. In *Proceedings of Information Hiding, First International Workshop*, Cambridge, UK, 1996.

[36] D. Kersten. Predictability and redundancy of natural images. *Journal of the Optical Society of America A*, 4(12):2395–2400, 1987.

[37] M. Kwan. *Gifshuffle*. `www.darkside.com.au/gifshuffle`.

[38] A. Latham. *Jpeg Hide-and-Seek*. `linux01.gwdg.de/alatham/stego`.

[39] D. Lowe. Distinctive image features from scale-invariant keypoints. *Internation Journal on Computer Vision*, 60(2):91–110, 2004.

[40] S. Lyu. Automatic image orientation determination with natural image statistics. Technical Report TR2005-545, Department of Computer Science, Dartmouth College, 2005.

[41] S. Lyu and H. Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[42] S. Lyu and H. Farid. Steganalysis using color wavelet statistics and one-class support vector machines. In *SPIE Symposium on Electronic Imaging*, San Jose, CA, 2004.

[43] A. B. M. B. M. Marcellin, M. Gormish. An overview of JPEG-2000. *Proceedings of IEEE Data Compression Conference*, pages 523–541, 2000.

[44] R. Machado. *EZStego*. `www.ezstego.com`.

[45] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern and Machine Intelligence*, 11:674–693, 1989.

[46] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (11):674–693, 1989.

[47] A. McNamara. Evaluating realism. In *Perceptually Adaptive Graphics, ACM SIGGRAPH and Eurographics Campfire*, Snowbird, Utah, 2001.

[48] G. W. Meyer, H. E. Rushmeier, M. F. Cohen, D. P. Greenberg, and K. E. Torrance. An experimental evaluation of computer graphics imagery. *ACM Transactions on Graphics*, 5(1):30–50, 1986.

[49] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using a generalized gaussian and complexity priors. *IEEE Transactions on Information Theory*, (45):909–919, 1999.

[50] J. Ogden, E. Adelson, J. Bergen, and P. Burt. Pyramid-based computer graphics. *RCA Engineer*, 30(5):4–15, 1985.

[51] N. M. Orenstein, editor. *Pieter Bruegel the Elder*. Yale University Press, New Haven and London, 2001.

[52] A. P. Pentland. Fractal based description of natural scenes. *IEEE Transactions on Pattern and Machine Intelligence*, 6(6):661–674, 1984.

[53] E. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding - a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[54] A. Popescu and H. Farid. Exposing digital forgeries by detecting traces of re-sampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005.

[55] A. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, in press, 2005.

[56] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int'l Journal of Computer Vision*, 40(1):49–71, December 2000.

[57] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int'l Journal of Computer Vision*, 40(1):49–71, October, 2000.

[58] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, November 2003.

[59] N. Provos and P. Honeyman. Detecting steganographic content on the internet. Technical Report CITI 01-1a, University of Michigan, 2001.

[60] N. Provos and P. Honeyman. Detecting steganographic content on the internet. In *ISOC NDSS'02*, San Diego, CA, 2002.

[61] P. M. Rademacher. *Measuring the Perceived Visual Realism of Images*. PhD thesis, UNC at Chapel Hill, 2002.

[62] D. L. Ruderman and W. Bialek. Statistics of natural image: Scaling in the woods. *Phys. Rev. Letters*, 73(6):814–817, 1994.

[63] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. In *Neural Computation*, pages 1443–1471, 2001.

[64] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge, 2004.

[65] E. Simoncelli. *Statistical Modeling of Photographic Images*, chapter 7. Handbook of Video and Image Processing. Academic Press, 2nd edition, 2005.

[66] E. P. Simoncelli. A rotation-invariant pattern signature. In *International Conference on Image Processing (ICIP)*, 1996.

[67] E. P. Simoncelli. Modeling the joint statistics of images in the wavelet domain. In *Proceedings of the 44th Annual Meeting*, volume 3813, Denver, CO, USA, 1999.

[68] E. P. Simoncelli and E. H. Adelson. *Subband image coding*, chapter Subband transforms, pages 143–192. Kluwer Academic Publishers, 1990.

[69] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Second Int'l Conf on Image Proc*, volume III, pages 444–447, Washington, DC, October 1995. IEEE Sig Proc Society.

[70] G. Strang. *Wavelet and filter banks*. Wiley, 2000.

[71] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *IEEE International Workshop on Content-based Access of Image and Video Databases, in conjunction with ICCV'98*, pages 42–51, Bombay, INDIA, 1998.

[72] R. Taylor, A. P. Micolich, and D. Jones. Fractal analysis of pollock's drip paintings. *Nature*, 399:422, 1999.

[73] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *IEEE Intl. Conference on Computer Vision (ICCV)*, Nice, France, 2003.

[74] A. Torralba and A. Oliva. Semantic organization of scenes using discriminant structural templates. In *International Conference on Computer Vision*, 1999.

[75] D. Upham. *Jsteg*. `ftp.funet.fi`.

[76] P. P. Vaidyanathan. Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques. *IEEE ASSP Magazine*, 4(3):4–20, 1987.

[77] A. Vailaya, A. K. Jain, and H.-J. Zhang. On image classification: City vs.landscapes. *International Journal of Pattern Recognition*, (31):1921–1936, 1998.

[78] V. Vapnik. *The nature of statistical learning theory*. Spring Verlag, 1995.

[79] M. Vetterli. A theory of multirate filter banks. *IEEE Transactions on ASSP*, 35(3):356–372, 1987.

[80] M. Wainwright and E. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In *Advances in Neural Information Processing Systems*, volume 12, pages 855–861. MIT Press, 2000.

[81] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 1991.

[82] A. Westfeld. *F5*. `wwwwrn.inf.tu-dresden.de/westfeld/f5`.

[83] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In *Proceedings of Information Hiding, Third International Workshop*, Dresden, Germany, 1999.

[84] X. Wu, S. Dumitrescu, and Z. Wang. Detection of lsb steganography via sample pair analysis. In *5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, 2002.

[85] S. C. Zhu, Y. Wu, and D. Mumford. Filters, random fields and maximum entropy (frame) - towards the unified theory for texture modeling. In *IEEE Conference Computer Vision and Pattern Recognition*, pages 686–693, San Francisco, CA, 1996.