1

Introduction

There are many daily pattern recognition tasks that humans routinely carry out without thinking twice. For example, we can recognize those that we know by looking at their face or hearing their voice. You can recognize the letters and words you are reading now because you have trained yourself to recognize English letters and words. We can understand what someone is saying even if it is slightly distorted (e.g., spoken too fast). However, human pattern recognition suffers from three main drawbacks: poor speed, difficulty in scaling, and inability to handle some recognition tasks. Not surprisingly, humans can't match machine speeds on pattern recognition tasks where good pattern recognition algorithms exist. Also, human pattern recognition ability gets overwhelmed if the number of classes to recognize becomes very large. Although humans have evolved to perform well on some recognition tasks such as face or voice recognition, except for a few trained experts, most humans cannot tell whose fingerprint they are looking at. Thus, there are many interesting pattern recognition tasks for which we need machines.

The field of machine learning or pattern recognition is rich with many elegant concepts and results. One set of pattern recognition methods that we feel has not been explained in sufficient detail is that of correlation filters. One reason why correlation filters have not been employed more for pattern recognition applications is that their use requires background in and familiarity with different disciplines such as linear systems, random processes, matrix/vector methods, statistical decision theory, pattern recognition, optical processing, and digital signal processing. This book is aimed at providing such background as well as introducing the reader to state-of-the-art in design and analysis of correlation filters for pattern recognition. The next two sections in this chapter will provide a brief introduction to pattern recognition and correlation, and in the last section we provide a brief outline of the rest of this book.

1.1 Pattern recognition

In pattern recognition, the main goal is to assign an observation into one of multiple classes. The observation can be a signal (e.g., speech signal), an image (e.g., an aerial view of a ground scene) or a higher-dimensional object (e.g., video sequence, hyperspectral signature, etc.) although we will use an image as the default object in this book. The classes depend on the application at hand. In automatic target recognition (ATR) applications, the goal may be to classify the input observation as either natural or man-made, and follow this up with finer classification such as vehicle vs. non-vehicle, tanks vs. trucks, one type of tank vs. another type.

Another important class of pattern recognition applications is the use of biometric signatures (e.g., face image, fingerprint image, iris image, and voice signals) for person identification. In some biometric recognition applications (e.g., accessing the automatic teller machine), we may be looking at a verification application where the goal is to see whether a stored template matches the live template in order to accept the subject as an authorized user. In other biometric recognition scenarios (e.g., deciding whether a particular person is in a database), we may want to match the live biometric to several stored biometric signatures.

One standard paradigm for pattern recognition is shown in Figure 1.1. The observed input image is first preprocessed. The goals of preprocessing depend very much on the details of the application at hand, but can include: reducing the noise, improving the contrast or dynamic range of the image, enhancing the edge information in the image, registering the image, and other application-specific processes.

A feature extraction module next extracts features from the preprocessed image. The goal of feature extraction is to produce a few descriptors to capture the essence of an input image. The number of features is usually much smaller than the number of pixels in that input image. For example, a 64×64 image contains 4096 numbers (namely the pixel values), yet we may be able to capture the essence of this image using only 10 or 20 features. Coming up with good features depends very much on the designer's experience in an application domain. For example, for fingerprint recognition, it is well known that features such as ridge endings and bifurcations called minutiae (shown in

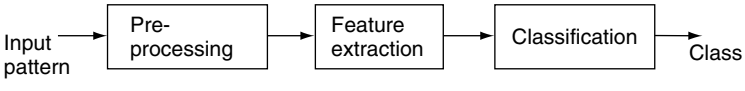

Figure 1.1 Block diagram showing the major steps in image pattern recognition

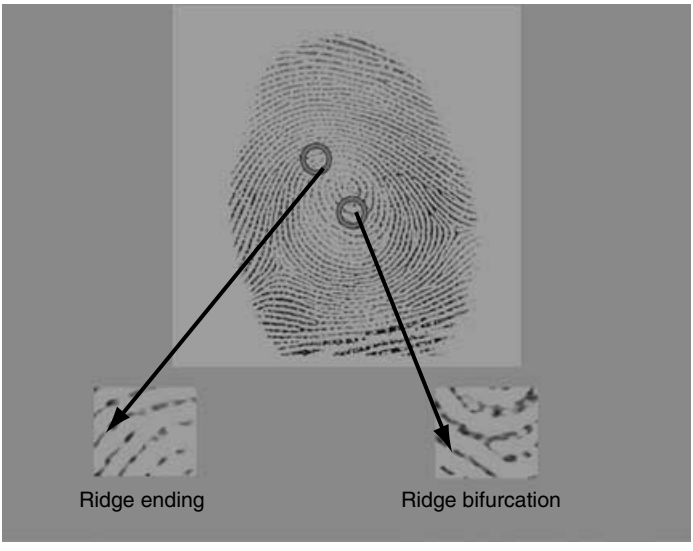

Figure 1.2 Some features used for fingerprint recognition: ridge ending (left) and ridge bifurcation (right)

Figure 1.2) are useful for distinguishing one fingerprint from another. In other pattern recognition applications, different features may be used. For example, in face recognition, one may use geometric features such as the distance between the eyes or intensity features such as the average gray scale in the image, etc. There is no set of features that is a universal set in that it is good for all pattern recognition problems. Almost always, it is the designer's experience, insight, and intuition that help in the identification of good features.

The features are next input to a classifier module. Its goal is to assign the features derived from the input observation to one of the classes. The classifiers are designed to optimize some metric such as probability of classification error (if underlying probability densities are known), or empirical error count (if a validation set of data with known ground truth[1] is available). Classifiers come in a variety of flavors including statistical classifiers, artificial neural-network-based classifiers and fuzzy logic-based classifiers. The suitability of a classifier scheme depends very much on the performance metric of interest, and on what a-priori information is available about how features appear for different classes. If we have probability density functions for various features for different classes, we can design statistical classification schemes. Sometimes, such probability density information may not be available and, instead, we may have sample feature vectors from different classes. In such a

[1] A term from remote sensing to denote the correct class of the object being tested.

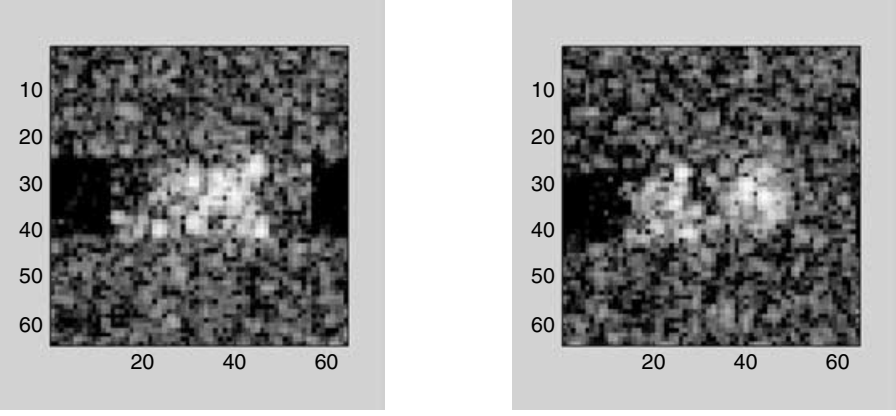

Figure 1.3 Synthetic aperture radar (SAR) images of two vehicles, (a) T72 and (b) BTR70, from the public MSTAR database [3]

situation, we may want to use trainable classifiers such as neural networks. In this book, we will not discuss these different pattern recognition paradigms. Interested readers are encouraged to consult some of the many excellent references [1, 2] discussing general pattern recognition methods.

Another important pattern recognition paradigm is to use the training data directly instead of first determining some features and performing classification based on those features. While feature extraction works well in many applications, it is not always easy for humans to identify what the good features may be. This is particularly difficult when we are facing classification problems such as the one shown in Figure 1.3, where the images were acquired using a synthetic aperture radar (SAR) and the goal is to assign the SAR images to one of two classes (tank vs. truck). Humans are ill equipped to come up with the "best" features for this classification problem. We may be better off letting the images speak for themselves, rather than imposing our judgments of what parts of SAR images are important and consistent in the way a target appears in the SAR imagery. Correlation pattern recognition (CPR) is an excellent paradigm for using training images to design a classifier and to classify a test image.

1.2 Correlation

Most readers are probably familiar with the basic concept of correlation as it arises in probability theory. We say that two random variables (RVs, the

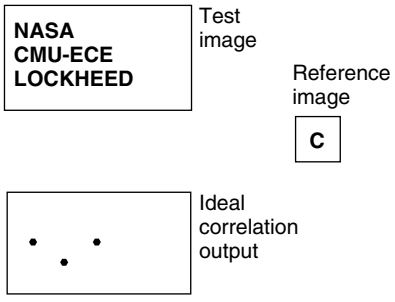

Figure 1.4 Schematic of the image correlation: reference image, test image, and ideal correlation output

concept to be explained more precisely in Chapter 2) are correlated if knowing something about one tells you something about the other RV. There are degrees of correlation and correlation can be positive or negative. The role of correlation for pattern recognition is not much different in that it tries to capture how similar or different a test object is from training objects. However, straightforward correlation works well only when the test object matches well with the training set and, in this book, we will provide many methods to improve the basic correlation and to achieve attributes such as tolerance to real-world differences or distortions (such as image rotations, scale changes, illumination variations, etc.), and discrimination from other classes.

We will introduce the concept of CPR using Figure 1.4. In this figure, we have two images: a reference image of the pattern we are looking for and a test image that contains many patterns. In this example, we are looking for the letter "C." But in other image recognition applications, the reference $r[m, n]$ can be an (optical, infrared, or SAR) image of a tank and the test image $t[m, n]$ can be an aerial view of the battlefield scene. In a biometric application, the reference may be a client's face image stored on a smart card, and the test image may be the one he is presenting live to a camera. For the particular case in Figure 1.4, let us assume that the images are binary with black regions taking on the value 1 and white regions taking on the value 0.

The correlation of the reference image $r[m, n]$ and the test image $t[m, n]$ proceeds as follows. Imagine overlaying the smaller reference image on top of the upper left corner portion of the test image. The two images are multiplied (pixel-wise) and the values in the resulting product array are summed to obtain the correlation value of the reference image with the test image for that relative location between the two. This calculation of correlation values is then repeated by shifting the reference image to all possible centerings of the reference image with respect to the test image. As indicated in the idealized

correlation output in Figure 1.4, large correlation values should be obtained at the three locations where the reference matches the test image. Thus, we can locate the targets of interest by examining the correlation output for peaks and determining if those correlation peaks are sufficiently large to indicate the presence of a reference object. Thus, when we refer to CPR in this book, we are not referring to just one correlation value (i.e., one inner product of two arrays), but rather to a correlation output $c[m, n]$ that can have as many pixels as the test image. The following equation captures the cross-correlation process

$$c[m,n] = \sum_k \sum_l t[k,l] r[k+m, l+n] \tag{1.1}$$

From Eq. (1.1), we see that correlation output $c[m, n]$ is the result of adding many values, or we can say that the correlation operation is an integrative operation. The advantage of such an integrative operation is that no single pixel in the test image by itself is critical to forming the correlation output. This results in the desired property that correlation offers graceful degradation. We illustrate the graceful degradation property in Figure 1.5. Part (a) of this figure shows a full face image from the Carnegie Mellon University (CMU) Pose, Illumination, and Expression (PIE) face database [4] and part (b) shows the correlation output (in an isometric view) from a CPR system designed to search for the image in part (a). As expected, the correlation output exhibits a large value indicating that the test image indeed matches the reference image. Part (c) shows the same face except that a portion of the face image is occluded. Although the resulting correlation output in part (d) exhibits correlation peaks smaller than in part (b), it is clear that a correlation peak is still present indicating that the test image does indeed match the reference object. Some other face recognition methods (that rely on locating both eyes to start the feature extraction process) will not exhibit similar graceful degradation properties.

Another important benefit of CPR is the in-built shift-invariance. As we will show in later chapters, correlation operation can be implemented as a linear, shift-invariant filter (this shift-invariance concept will be made more precise in Chapter 3 on linear systems), which means that if the test image contains the reference object at a shifted location, the correlation output is also shifted by exactly the same amount. This shift-invariance property is illustrated in parts (e) and (f) of Figure 1.5. Part (e) shows a shifted and occluded version of the reference image and the resulting correlation output in part (f) is shifted by the same amount, but the correlation peak is still very discernible. Thus, there is no need to go through the trouble of centering the input image prior to recognizing it.

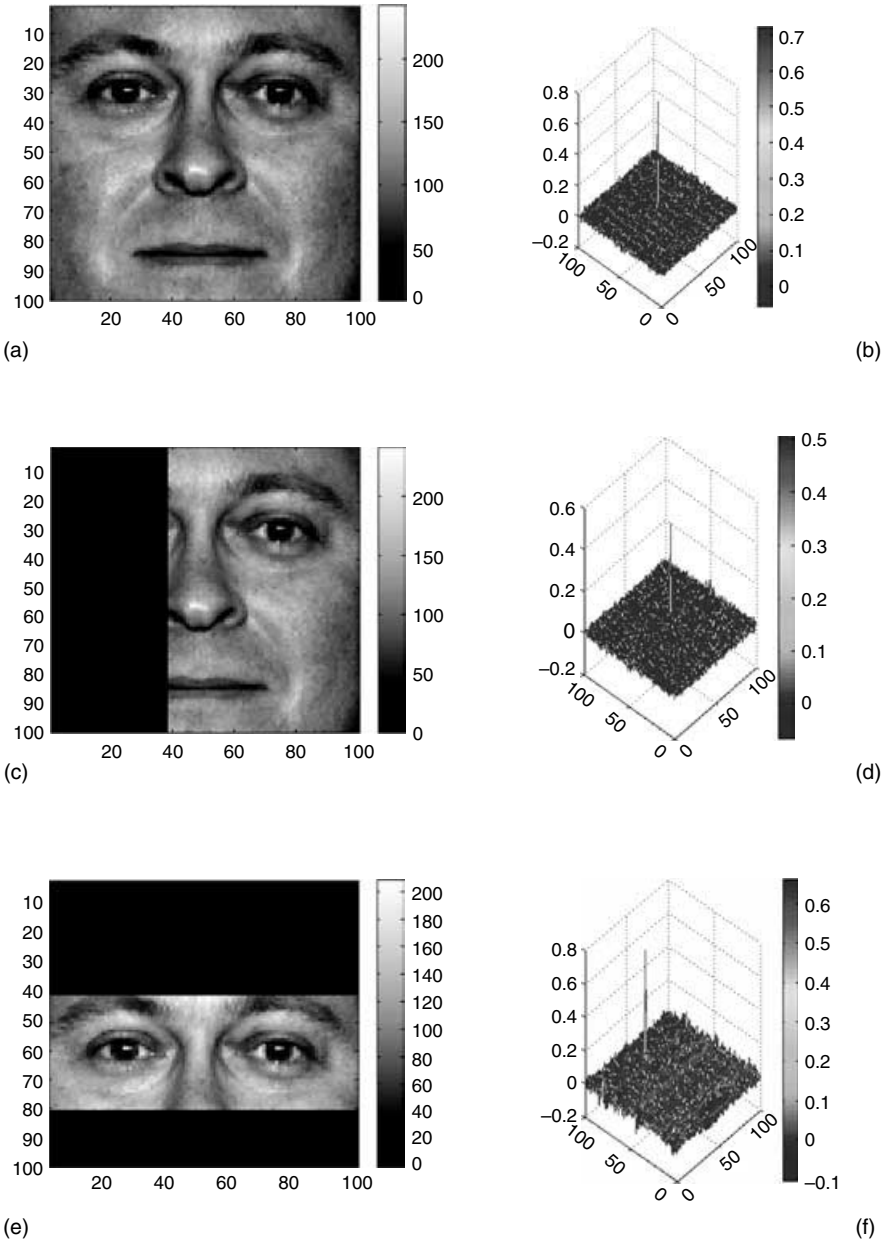

Figure 1.5 Illustration of the graceful degradation property of correlation operation, (a) a full face image from the CMU PIE database [4], (b) correlation output for test image in part (a), (c) occluded face image, (d) correlation output for image in part (c), (e) shifted and occluded face image, and (f) correlation output for image in part (e)

A reasonable question to ask at this stage is why one needs to read the rest of this book when we have already explained using Figure 1.4 and Figure 1.5 the basic concept of correlation and advantages of using correlation.

We need to discuss more advanced correlation filters because the simple scheme in Figure 1.4 works well only if the test scene contains exact replicas of the reference images, and if there are no other objects whose appearance is similar to that of the reference image. For example, in Figure 1.4, the letter "O" will be highly correlated with letter "C" and the simple cross-correlation will lead to a large correlation output for the letter "O" also, which is undesirable. Thus, we need to, and we will, discuss the design of correlation templates that not only recognize the selected reference image, but also reject impostors from other classes. Also the book discusses practical issues of computing correlation using digital methods and optical methods. One way to summarize the contents of this book is that it contains much of the material we wish had been available when starting into CPR.

Another deficiency of the straightforward correlation operation in Eq. (1.1) is that it can be overly sensitive to noise. Most test scenes will contain all types of noise causing randomness in the correlation output. If this randomness is not explicitly dealt with, correlation outputs can lead to erroneous decisions. Also, as illustrated in Figure 1.5, sharp correlation peaks are important in estimating the location of a reference image in the test scene. It is easier to locate the targets in a scene if the correlation template is designed to produce sharp peaks. Unfortunately, noise tolerance and peak sharpness are typically conflicting criteria, and we will need design techniques that optimally trade off between the two conflicting criteria.

The straightforward correlation scheme of Figure 1.4 does not work well if the reference image appears in the target scene with significant changes in appearance (often called *distortions*), perhaps owing to illumination changes, viewing geometry changes (e.g., rotations, scale changes, etc.). For example, a face may be presented to a face verification system in a different pose from the one used at the time of enrolment. In an ATR example based on infrared images, a vehicle of interest may look different when compared to the reference image because the vehicle may have been driven around (and as a result, the engine has become hot leading to a brighter infrared image). A good recognition system must be able to cope with such expected variability. In this book, we will discuss various ways to increase the capabilities of correlation methods to provide distortion-tolerant pattern recognition.

Another important question in connection with the correlation method is how it should be implemented. As we will show later in this book, straightforward implementations (e.g., image–domain correlations as in Figure 1.4) are

inefficient, and more efficient methods based on fast Fourier transforms (FFTs) exist. Such efficiency is not just a theoretical curiosity; this efficiency of FFT-based correlations is what allows us to use CPR for demanding applications such as real-time ATR and real-time biometric recognition. This book will provide the theory and details to achieve such efficiencies.

It is fair to say that the interest in CPR is mainly due to the pioneering work by VanderLugt [5] that showed how the correlation operation can be implemented using a coherent optical system. Such an optical implementation carries out image correlations "at the speed of light." However, in practice, we don't achieve such speed owing to a variety of factors. For example, bringing the test images and reference images into the optical correlators and transferring the correlation outputs from the optical correlators for post-processing prove to be bottlenecks, as these steps involve conversion from electrons to photons and vice versa. Another challenge is that the optical devices used to represent the correlation templates cannot accommodate arbitrary complex values as digital computers can. Some optical devices may be phase-only (i.e., magnitude must equal 1), binary phase-only (i.e., only $+1$ and -1 values are allowed), or cross-coupled where the device can accommodate only a curvilinear subset of magnitude and phase values from the complex plane. It is necessary to design CPR schemes that take into account such implementation constraints if we want to achieve the best possible performance. This book will provide sufficient information for designing *optical* CPR schemes.

1.3 Organization

As discussed in the previous section, CPR is a rather broad topic requiring background in many subjects including linear systems, matrix and vector methods, RVs and processes, statistical hypothesis testing, optical processing, digital signal processing, and, of course, pattern recognition theory. Not surprisingly, it is difficult to find all these in one source. It is our goal to provide the necessary background in these areas and to illustrate how to synthesize that knowledge to design CPR systems. In what follows, we will provide brief summaries of what to expect in the following chapters.

Chapter 2, Mathematical background In this chapter, we provide brief reviews of several relevant topics from mathematics. We first review matrices and vectors, as the correlation templates (also known as *correlation filters*) are designed using linear algebra methods and it is important to know concepts such as matrix inverse, determinant, rank, eigenvectors, diagonalization, etc. This chapter also introduces some vector calculus (e.g., gradient) and

illustrates its use in optimization problems that we will need to solve for CPR. As we mentioned earlier, randomness is inevitable in input patterns, and a short review of probability theory and RVs is provided in this chapter. This review includes the case of two RVs as well as more RVs (equivalently, more compactly represented as a random vector).

Chapter 3, Linear systems and filtering theory In this chapter, we review the basic concepts of linear shift-invariant systems and filters. These are important for CPR since most implementations of correlation are in the form of filters, which is why we refer to the correlation templates also as correlation filters (strictly speaking, *templates* refer to image domain quantities whereas *filters* are in the frequency domain). In addition to standard one-dimensional (1-D) signals and systems topics, we review some two-dimensional (2-D) topics of relevance when dealing with images. This is the chapter where we will see that the correlation operation is implemented more efficiently via the frequency domain rather than directly in the image domain. Both optical and digital correlation implementations originate from this frequency domain version. This chapter reviews sampling theory, which is important to understand the connections between digital simulations and optical implementations. Since digital correlators are heavily dependent on the FFT, this chapter reviews the basics of both 1-D and 2-D FFTs. Finally, we review random signal processing, as the randomness in the test images is not limited to just one value or pixel. The randomness in the images may be correlated from pixel to pixel necessitating concepts from random processes, which are reviewed in this chapter.

Chapter 4, Detection and estimation The goal of this relatively short chapter is to provide the statistical basis for some commonly used CPR approaches. First, we derive the optimal methods for classifying an observation into one of two classes. Then, we show that the optimum method is indeed a correlator, if we can assume some conditions about the noise. Another topic of importance is estimation, which deals with the best ways to extract unknown information from noisy observations. This is of particular importance when we need to estimate the error rates from a correlator.

Chapter 5, Correlation filter basics In some ways, this is the core of this book. It starts by showing how correlation is optimum for detecting a known reference signal in additive white Gaussian noise (AWGN). This theory owes its origins to the matched filter (MF) [6], introduced during World War II for radar applications. Next, we show how MFs can be implemented digitally and optically using Fourier transforms (FTs). As MFs cannot be implemented (as they are) on limited-modulation optical devices, we next discuss several variants of the MF including phase-only filters, binary phase-only filters and