Automatic Target Recognition

Fourth Edition

Bruce J. Schachter

Tutorial Texts in Optical Engineering
Volume TT120

SPIE PRESS
Bellingham, Washington USA

Library of Congress Cataloging-in-Publication Data

Names: Schachter, Bruce J. (Bruce Jay), 1946– author.
Title: Automatic target recognition / Bruce J. Schachter.
Description: Fourth edition. | Bellingham, Washington : SPIE Press, [2020] | Series: Tutorial texts in optical engineering ; volume TT120 | Includes bibliographical references and index.
Identifiers: LCCN 2019048700 | ISBN 9781510631199 (paperback) | ISBN 9781510631205 (pdf) | ISBN 9781510631229 (kindle edition) | ISBN 9781510631212 (epub)
Subjects: LCSH: Radar targets. | Optical pattern recognition. | Algorithms. | Image processing.
Classification: LCC TK6580 .S33 2020 | DDC 623.7/348–dc23
LC record available at https://lccn.loc.gov/2019048700

Published by

SPIE
P.O. Box 10
Bellingham, Washington 98227-0010 USA
Phone: +1 360.676.3290
Fax: +1 360.647.1445
Email: books@spie.org
Web: http://spie.org

Printed in the United States of America.
First Printing.
For updates to this book, visit http://spie.org and type "TT120" in the search field.

SPIE.

Contents

Preface

An automatic target recognizer (ATR) is a real-time or near-real-time image/signal-understanding system. An ATR is presented with a stream of data. It outputs a list of the targets that it has detected and recognized in the data provided to it. A complete ATR system can also perform other functions such as image stabilization, preprocessing, mosaicking, target tracking, activity recognition, multi-sensor fusion, sensor/platform control, and data packaging for transmission or display.

In the early days of ATR, there were fierce debates between proponents of signal processing and those in the emerging field of computer vision. Signal processing fans were focused on more advanced correlation filters, stochastic analysis, estimation and optimization, transform theory, and time-frequency analysis of nonstationary signals. Advocates of computer vision said that signal processing provides some nice tools for our toolbox, but what we really want is an ATR that works as well as biological vision. ATR designers were less interested in processing signals than understanding scenes. They proposed attacking the ATR problem through artificial intelligence (AI), computational neuroscience, evolutionary algorithms, case-based reasoning, expert systems, and the like. Signal processing experts are interested in tracking point-like targets. ATR engineers want to track a target with some substance to it, identify what it is, and determine what activity it is engaged in. Signal processing experts keep coming up with better ways to compress video. ATR engineers want more intelligent compression. They want the ATR to tell the compression algorithm which parts of the scene are more important and hence deserving of more bits in the allocation. ATR, in and of itself, can be thought of as a data reduction technique. The ATR takes in a lot of data and outputs relatively little data. Data reduction is necessary due to bandwidth limitations of the data link and workload limits of the time-strapped human operator. People are very good at analyzing video until fatigue sets in or they get distracted. They don't want to be like the triage doctor at the emergency ward, assessing everything that comes in the door, continually assigning priorities to items deserving further attention. Pilots and ground station operators want a machine to relieve their burden as long as it rarely makes a mistake. Trying to do this keeps ATR engineers employed. As often told to the author, pilots and image analysts are not looking for machines to replace them entirely. However, such decisions will be made higher up in the chain of command as ATR technology progresses.

The human vision system is not "designed" to analyze certain kinds of data such as rapid step-stare imagery, complex-valued signals that arise in radars, hyperspectral imagery, 3D LADAR data, or fusion of signal data with various forms of precise metadata. ATR shines when the sustained data rate is too high or too prolonged for the human brain, or the data is not well suited for presentation to humans. Nevertheless, most current ATRs operate with humans-in-the-loop. Humans, at present, are much better than ATRs at tasks requiring consultation, comprehension, and judgement. Humans still make the final decision and determine the action to be taken. This means that ATR output, which is statistical and multi-faceted by nature, has to be presented to the human decision makers in an easily understood form. This is a difficult man–machine interface problem. Marching toward the future, more autonomous robotic systems will necessarily rely more on ATRs to substitute for human operators, possibly serving as the "brains" of entire robotic platforms. We leave this provocative topic to the end of the book.

Systems engineers took notice once ATRs became deployable. Systems engineers are grounded in harsh reality. They care little about the debate between signal processing and computer vision. They don't want to hear about an ATR being brain-like. They are not interested in which classification paradigm performs 1% better than the next. They care about the concept of operations (ConOps) and how it directs performance and functionality. They care about mission objectives and mission requirements. They want to identify all possible stakeholders, form an integrated product team, determine key performance parameters (KPPs), and develop test and evaluation (T&E) procedures to determine if performance requirements are met. Self-test is the norm for published papers and conference talks. Independent test and evaluation, laboratory blind tests, field tests, and software regression tests are the norm for determining if a system is deployable. The systems engineer's focus is broader than ATR performance. Systems engineers want the entire system, or system of systems, to work well, including platform, sensors, ATR, and data links. They want to know what data can be provided to the ATR and what data the ATR can provide to the rest of the system. They want to know how one part of the system affects all other parts of the system. Systems designers care a lot about size, weight, power, latency, current and future costs, logistics, timelines, mean time between failure, and product repair and upgrade. They want to know the implications of system capture by the enemy.

At one time, ATR was the sole charge of the large defense electronics companies, working closely with the Government labs. Only the defense companies and Government have fleets of data collection aircraft, high-end sensors, and access to foreign military targets. Although air-to-ground has been the focus of much ATR work, ATR actually covers a wide range of sensors, operating within or between the layers of space, air, ocean/land surface, and undersea/underground. Although the name ATR implies recognition of targets, ATR engineers have broader interests. ATR groups tackle any type of military problem involving the smart processing of imagery or signals. The Government (or Government-funded prime contractor) is virtually the only customer. So, some of the ATR engineer's time is spent reporting to the Government, participating in joint data collections, taking part in Government-sponsored tests, and proposing new programs to the Government.

Since the 1960s, the field of ATR has advanced in parallel with similar work in the commercial sector and academia, involving industrial automation, medical imaging, surveillance and security, video analytics, and space-based imaging. Technologies of interest to both the commercial and defense sector include low-power processors, novel sensors, increased system autonomy, people detection, robotics, rapid search of vast amounts of data (big data), undersea inspection, and remote medical diagnosis. The bulk of funding in some of these areas has recently shifted from the defense to the commercial sector. More money is spent on computer animation for Hollywood movies than for the synthesis of forward-looking infrared (FLIR) and synthetic aperture radar (SAR) imagery. The search engine companies are investing much more in neural networks compared to the defense companies. Well-funded brain research programs are investigating the very basis of human vision and cognitive processing. The days of specialized military processors (e.g., VHSIC) are largely over. Reliance is now on chips in high-volume production: multi-core processors (e.g., Intel and ARM), FPGAs (e.g., Xilinx and Intel/Altera), and GPUs (e.g., Nvidia and AMD). Highly packaged sensors (visible, FLIR, LADAR, and radar) combined with massively parallel processors are advancing rapidly for the automotive industry to meet new safety standards (e.g., Intel/MobilEye). Millions of systems will soon be produced per year. Current advanced driver assistance systems (ADAS) can detect pedestrians, animals, bicyclists, road signs, traffic lights, cars, trucks, and road markers. These are a lot like ATR tasks. The rapid advancement of ADAS will lead to driverless cars.

Some important differences between ATRs and commercial systems are worth noting. ATRs generally have to detect and recognize objects at much longer ranges than commercial systems. Enemy detection and recognition are non-cooperative processes. Although a future car might have a LADAR, radar, or FLIR sensor, it won't have one that can produce high-quality data from a 20,000-ft range. An ADAS will detect a pedestrian but won't report if he is carrying a rifle. Search engine companies need to search large volumes of data with an image-based search, but they don't have the metadata to help the search, such as is available on military platforms. That being said, the cost and innovation rate of commercial electronics can't be matched by military systems. The distinction between commercial and military systems is starting to blur in some instances. Cell phones now include cameras, inertial measurement units, GPS, computers, algorithms, and transmitters/receivers. Slightly rugged versions of commercial cell phones and tablet computers are starting to be used by the military, even with ATR apps. “Toy” drones are approaching the sophistication of the smallest military unmanned air vehicles. They are now produced in volumes of a million per year. ATR engineers are in tune with advances in the commercial sector and their applicability to ATR. Even their hobbies tend to focus on technology, e.g., hobbies such as quadcopters, novel cameras, 3D printers, computers, phone apps, robots, etc.

ATR is not limited to a device; it is also a field of research and development. ATR technology can be incorporated into systems in the form of self-contained hardware, FPGA code, or higher-level language code. ATR groups can help add autonomy to many types of systems. ATR can be viewed very narrowly or very broadly, borrowing concepts from a wide variety of fields. Papers on ATR are often of the form: “Automatic Target Recognition using XXX,” where the XXX can be any technology such as super-resolution, principal component analysis, sparse coding, singular value decomposition, Eigen templates, correlation filters, kinematic priors, adaptive boosting, hyperdimensional manifolds, Hough transforms, foveation, etc. In the more ambitious papers, the XXX is a mélange of technologies, such as fuzzy-rule-based expert systems, wavelet neural genetic networks, fuzzy morphological associative memory, optical holography, deformable wavelet templates, hierarchical support vector machines, Bayesian recognition by parts, etc. Get the picture? Nearly any type of technology, everything but the kitchen sink, can be thrown at the ATR problem, with scant large-scale independent competitive test results to indicate which approach really works best, supposing that “best” can be defined and measured. This book is not a comprehensive survey of every technology that has ever been applied to ATR. This book covers some of the basics of ATR. While some of the topics in this book can be found in textbooks on pattern recognition and computer vision, this book focuses on their application to military problems as well as the unique requirements of military systems.

The topics covered in the book are organized in the way one would design an ATR. The first step is to understand the military problem and make a list of potential solutions to the problem. A key issue is the availability of sufficiently comprehensive sets of data to train and test the potential solutions. This involves developing a sound test plan, specifying procedures and equations, and determining who is going to do the testing. Testing isn’t open ended. Exit criteria are needed to determine when a given test activity has been successfully completed. The next steps in ATR design are choosing the detector and classifier. The detector focuses attention on the regions-of-interest in the imagery requiring additional scrutiny. The classifier further processes these regions-of-interest and is the decision engine for class assignment. It can operate at any or all levels of a decision tree, from clutter rejection to identifying a specific vehicle or activity. Detected targets are often tracked. Target tracking has historically been treated as a separate subject from ATR, mainly because point-like targets contain too little information to apply an ATR. However, as sensor resolution improves, the engineering disciplines of target tracking and ATR are starting to merge. The ATR and tracker can be united for efficiency and performance. The fifth chapter covers the basics of multisensory fusion. Then it broadens the topic to a variety of other forms of fusion. A strawman design is provided for a more advanced ATR, but with no claim that this is the only way to construct a next-generation ATR. The strawman design should be thought of as a brainstormed simple draft proposal intended to generate discussion of its advantages and disadvantages, and to trigger the generation of new and better proposals. Future ATRs will have to combine data from multiple sources. The seventh chapter points out how primitive current ATRs really are, as compared to biological systems. It suggests ways for measuring the intelligence of an ATR. This goes far beyond the basic performance measurement techniques covered in Chapter 1. The final chapter examines the role of ATR in its ultimate embodiment—that being lethal autonomous robots. These are air, land, or sea weapons that detect, track, recognize, and attack targets on their own. There is no human-in-the-loop to control the attack; instead, the weapon itself decides when and what to strike, based on guidelines provided to it. Such weapons can come in the form of unmanned ground vehicles, unmanned undersea vehicles, or swarms of mini-drones. The chapter covers legal, moral, ethical, and technical issues, as well as what can go wrong. The first appendix lists the many resources available to the ATR engineer. Many of the listed agencies supply training and testing data, perform blind tests, and sponsor research into compelling new sensor and ATR designs. The second appendix advances the notion that a problem that is well described is half solved. The third appendix explains the acronyms and abbreviations used in the book.

CHAPTER 1: ATR technology has benefited from a significant investment over the last 50 years. However, the once-accepted definitions and evaluation criteria have been displaced by the march of technology. The first chapter updates the language for describing ATR systems and provides well-defined criteria for evaluating such systems. This will advance collaboration between ATR developers, evaluators, and end-users.

ATR is used as an umbrella term for a broad range of military technology beyond just the recognition of targets. In a more general sense, ATR means *sensor data exploitation*. Two types of definitions are included in the first chapter. One type defines fundamental concepts. The other type defines basic performance measures. In some cases, definitions consist of a list of alternatives. This approach enables choices to be made to meet the needs of particular programs. The important point to keep in mind is that within the context of a particular experimental design, a set of protocols should be adopted to best fit the situation, applied, and then kept constant throughout the evaluation. This is especially important for competitive testing.

The definitions given in Chapter 1 are intended for evaluation of end-to-end ATR systems as well as the prescreening and classifier stages of the systems. Sensor performance and platform characteristics are excluded from the evaluation. It is recognized that sensor characteristics and other operational factors affect the imagery and associated metadata. A thorough understanding of data quality, integrity, synchrony, availability, and timeline are important for ATR development, test, and evaluation. Data quality should be quantified and assessed. However, methods for doing so are not covered in this book. The results and validity of ATR evaluation depend on the representativeness and comprehensiveness of the development and test data. The adequacy of development and test data is primarily a budgetary issue. The ATR engineer should understand and be able to convey the implications of limited, surrogate, or synthetic data. The ATR engineer should be able to damp down naïve proposals centered around the use of an off-the-shelf deep-learning neural network as a miraculous cure to the alleged ATR affliction.

Chapter 1 formalizes definitions and performance measures associated with ATR evaluation. All performance measures must be accepted as ballpark predictions of actual performance in combat. More carefully formulated experiments will provide more meaningful conclusions. The final measure of effectiveness takes place in the battlefield.

CHAPTER 2: Hundreds of simple target detection algorithms were tested on mid- and longwave FLIR images, as well as X-band and Ku-band SAR images. Each algorithm is briefly described. Indications are given as to which performed well. Some of these simple algorithms are loosely derived from standard tests of the difference of two populations. For target detection, these are typically populations of pixel grayscale values or features derived from them. The statistical tests are often implemented in the form of sliding triple-window filters. Several more-elaborate algorithms are also described with their relative performances noted. These algorithms utilize neural networks, deformable templates, and adaptive filtering. Algorithm design issues are broadened to cover system design issues and concepts of operation.

Since target detection is such a fundamental problem, it is often used as a test case for developing technology. New technology leads to innovative approaches for attacking the problem. Eight inventive paradigms, each with deep philosophical underpinnings, are described in relation to their effect on target detector design.

CHAPTER 3: Target classification algorithms have generally kept pace with developments in the academic and commercial sectors since the 1970s. However, most recently, investment into object classification by Internet companies and various large-scale projects for understanding the human brain has far outpaced that of the defense sector. The implications are noteworthy.

There are some unique characteristics of the military classification problem. Target classification is not solely an algorithm design problem, but is part of a larger system design task. The design flows down from a ConOps and KPPs. Required classification level is specified by contract. Inputs are image and/or signal data and time-synchronized metadata. The operation is often real-time. The implementation minimizes size, weight, and power (SWaP). The output must be conveyed to a time-strapped operator who understands the rules of engagement. It is assumed that the adversary is actively trying to defeat recognition. The target list is often mission dependent, not necessarily a closed set, and can change on a daily basis. It is highly desirable to obtain sufficiently comprehensive training and testing data sets, but costs of doing so are very high, and data on certain target types are scarce or nonexistent. The training data might not be representative of battlefield conditions, suggesting the avoidance of designs tuned to a narrow set of circumstances. A number of traditional and emerging feature extraction and target classification strategies are reviewed in the context of the military target classification problem.

CHAPTER 4: The subject being addressed is how an automatic target tracker (ATT) and an ATR can be fused so tightly and so well that their distinctiveness becomes lost in the merger. This has historically not been the case outside of biology and a few academic papers. The biological model of ATT∪ATR arises from dynamic patterns of activity distributed across many neural circuits and structures (including those in the retinae). The information that the brain receives from the eyes is “old news” at the time that it receives it. The eyes and brain forecast a tracked object’s future position, rather than relying on the perceived retinal position. Anticipation of the next moment—building up a consistent perception—is accomplished under difficult conditions: motion (eyes, head, body, scene background, target) and processing limitations (neural noise, delays, eye jitter, distractions). Not only does the human vision system surmount these problems, but it has innate mechanisms to exploit motion in support of target detection and classification. Biological vision doesn’t normally operate on snapshots. Feature extraction, detection, and recognition are spatiotemporal. When scene understanding is viewed as a spatiotemporal process, target detection, target recognition, target tracking, event detection, and activity recognition (AR) do not seem as distinct as they are in current ATT and ATR designs. They appear as similar mechanisms taking place at varying time scales. A framework is provided for unifying ATT, ATR, and AR.

CHAPTER 5: Predatory animals detect, stalk, recognize, track, chase, home in on, and if lucky, catch their prey. Stereo vision is generally their most important sensor asset. Most predators also have a good sense of hearing. Some predators can smell their prey from a mile away. Most creatures combine data from multiple sensors to eat or avoid being eaten. Different creatures use different combinations of sensors, including sensors that detect vibration, infrared radiation, various spectral bands, polarization, Doppler, and magnetism. Biomimicry suggests that a combination of diverse sensors works better than use of a single sensor type. Sensor fusion intelligently combines sensor data from disparate sources such that the resulting information is in some ways superior to the data from a single source. Chapter 5 provides techniques for low-level, mid-level, and high-level information fusion. Other forms of fusion are also of interest to the ATR engineer. Multifunction fusion combines functions normally implemented by separate systems into a single system. Zero-shot learning (ZSL) is a way of recognizing a target without having trained on examples of the target. ZSL provides a vivid description of a detected target as a fusion of its semantic attributes. The commercial world is embracing multisensor fusion for driverless cars. New sensor and processor designs are emerging with applicability to autonomous military vehicles.

CHAPTER 6: Traditional feedforward neural networks, including multilayer perceptrons (MLPs) and the newly popular convolutional neural networks (CNNs), are trained to compute a function that maps an input vector to an output vector. The N-element output vector can convey estimates of the probabilities of N target classes. Nearly all current ATRs perform target classification using feedforward neural networks. These can be shallow or deep. The ATR detects a candidate target, transforms it to a feature vector, and then processes the vector unidirectionally, step by step; the number of steps is proportional to the number of layers in the neural network. Signals travel one way from input to output. A recurrent neural network (RNN) is an appealing alternative. Its neurons send feedback signals to each other. These feedback loops allow RNNs to exhibit dynamic temporal behavior. The feedback loops also establish a type of internal memory. While feedforward neural networks are generally trained in a supervised fashion by backpropagation of output error, RNNs are trained by backpropagation through time.

Although feedforward neural networks are said to be inspired by the architecture of the brain, they do not model many abilities of the brain, such as natural language processing and visual processing of spatiotemporal data. Feedback is omnipresent in the brain, endowing both short-term and long-term memory. The human brain is thus an RNN—a network of neurons with feedback connections. It is a dynamical system. The brain is plastic, adapting to the current situation. The human vision system not only learns patterns in sequential data, but even processes still frame (snapshot) data quite well with its RNN, jerking the eyes in saccades to shift focus over key points on a snapshot, turning the snapshot into a movie.

An improved type of RNN, called long short-term memory (LSTM), was developed in the 1990s by Jürgen Schmidhuber and his former Ph.D. student Sepp Hochreiter. LSTM and its many variants are now the predominant RNN. LSTM is said to be in use in billions of commercial devices.

Brains don't come in a box like a desktop computer or supercomputer. All natural intelligence is embodied and situated. Many military systems, such as unmanned air vehicles and robot ground vehicles, are embodied and situated. The body (platform) maneuvers the sensor systems to view the battlespace from different situations. An ATR based on an RNN, that is embodied and situated [ES], adaptive and plastic [Pl], and of limited precision (e.g., 16-bit floating point), will be denoted by the model $\mathbf{M} = \text{ES-Pl-RNN}(\mathbb{Q}_{16})$. A recurrent ATR is more powerful in many ways than a standard ATR. Both computationally more powerful and biologically more plausible than other types of ATRs, an RNN-based ATR understands the notion of events that unfold over time. Its design can benefit from ongoing advances in neuroscience.

Professor Schmidhuber has made an additional improvement to his model. He tightly couples a controller **C** to a model **M**. Both can be RNNs or composite designs incorporating RNNs. Following Schmidhuber's lead, we propose a strawman ATR that couples a controller **C** to our model $\mathbf{M}=\text{ES-Pl-RNN}(\mathbb{Q}_{16})$ to form a complete system ($\mathbf{C} \cup \mathbf{M}$) that is more powerful in many ways than a standard ATR. $\mathbf{C} \cup \mathbf{M}$ can learn a never-ending sequence of tasks, operate in unknown environments, realize abstract planning and reasoning, perform experiments, and retrain itself on-the-fly. This next-generation ATR is suitable for implementation on two chips: a single custom low-power chip (<1 W) for effecting **M**, hosted by a standard processor serving as the controller **C**. A heterogeneous chip design incorporating high-speed I/O, multicore ARM processors, logic gates, GPU, codec, and neural section is also appropriate. This next-generation ATR is applicable to various military systems, including those with extreme size, weight, and power constraints.

CHAPTER 7: ATRs have been under development since the 1960s. Advances in computer processing, computer memory, and sensor resolution are easy to evaluate. However, the time horizon of the truly smart ATR seems to be receding at a rate of one year per year. One issue is that there has never been a way to measure the intelligence of an ATR. This is fundamentally different from measuring detection and classification performance. The description of what constitutes an ATR, and in particular a smart ATR, keeps changing. Early ATRs did little more than detect fuzzy bright spots in first-generation

FLIR video or ten-foot-resolution SAR data. Sensors are getting better, computers are getting faster, and the ATR is expected to take over more of the workload. With unmanned systems there is no human onboard to digest information. The ATR is compelled to transmit only the most important information over a limited-bandwidth data link. The ATR or robotic system can be viewed as a substitute for a human. What constitutes intelligence in artificial humans has long been debated, starting with stories of golems, continuing to the Turing test, and including current dire predictions of super-intelligent robots superseding humans. Chapter 7 provides a Turing-like test for judging the intelligence of an ATR.

CHAPTER 8: Automation has advanced unceasingly for hundreds of years. The final chapter of this book reflects on the clash of automation and human values. At the forefront of this clash is the Lethal Autonomous Robot (LAR). LARs are defined as mobile, fully autonomous, offensive mechanized platforms that adapt their behavior to meet prescribed goals within a constantly changing environment. LARs are intelligent machines that detect, classify, track and kill their targets without human intervention. ATR is the essential component of the LAR. Unlike existing ATRs, which are generally just aided target recognizers, humans will be out of the immediate kill chain in LARs. The machine will determine what gets destroyed, and who gets killed, according the target list and rules of engagement provided to it. This places a heavy burden on the ATR to distinguish between combatants and noncombatants, military and civilian objects. However, even a smart ATR combined with smart decision-making software is deemed intolerable by many human rights organizations and some nation states. They are demanding that a human be in the loop with final control over an attack. They want to ban fully autonomous LARs altogether (or under the majority of circumstances), as has been done with chemical weapons, blinding lasers, cluster munitions, napalm, biological weapons and conventional anti-personnel mines.

However, the borderline between LARs and weapons controlled by humans is indistinct. If the ATR in a LAR can detect and recognize targets better than the humans in the loop, the humans will inevitably defer to the targeting decisions of the ATR. Then the humans will just be in the loop for ethical cover. But, the humans will lengthen the kill chain, making systems under human control slower and less effective than true LARs. Robotic aircraft, ground and undersea vehicles, and swarms of small craft of all types will be so cheap and so effective, that giving them more autonomy—including to kill—may well prove irresistible to nation states and non-state actors. This will be particularly true when one's powerful adversaries are relying on them. This is the real crux of the issue. As many countries are now racing to develop LARs, ethical issues will crash into realpolitik in the coming years.

Humans are better suited for leadership and command than robots. Robots are better suited for quick reaction and operation in dangerous situations like enemy fire, minefields, radioactive contamination and chemical attack. Human lives are precious. Robot "lives" not so much. Thus far, in the history of mankind, nothing has stopped automation. Nevertheless, robots will not replace humans in the larger loop of engineering and design, negotiating and treaty making, voting and political decisions, command and control. Robotic overlords subjugating humankind remain the stuff of science fiction and doomsayers.

LARs will be considered a success by some if their decisions to engage or not engage, kill or not kill, are speedier, and in some sense, superior to that of humans. Taking this several steps further, *for LARs to be revolutionary war machines*, the ATR at the heart of leading-edge LARs will need to be "brainier" than current ATRs in all the ways covered in Chapter 7. They will have to be able to operate alone or as part of human/robot teams. *LARs* will need to be easily understood and trusted by humans. They will have to analyze and explain their observations and how these observations led to their actions. LARs will need to draw rational conclusions (deduction), make plausible assumptions (abduction), and generalize from observations (induction). To reach this point, a LAR will require sufficient background knowledge, experience, adaptability, discernment and statistical thinking to turn incoming data into actions. All of this will be difficult to achieve. AI, neural networks, and ATR are often marketed as brain-like. However, no one knows enough about the brain to reverse engineer neural functioning. Beyond general human intelligence lies super-intelligence. LARs, let alone super-intelligent LARs, do not yet exist in any meaningful sense. Rudimentary LARs are under development for narrowly defined conditions and missions. But, LARs as smart and as capable, or smarter and more capable, than a well-trained soldier, sailor or pilot, are not imminent.

APPENDIX 1: The first appendix lists the many resources available to the ATR engineer and includes a brief historical overview of the technologies involved in ATR development.

APPENDIX 2: A successful project starts with a clear description of the problem to be solved. However, a well-defined ATR problem is surprisingly hard to come by. The second appendix provides some questions to pose to a customer to help get a project going.

APPENDIX 3: The third appendix defines all of the acronyms and abbreviations used in this book.

Acknowledgments

Special thanks to the United States Army Night Vision and Electronic Sensors Directorate (NVESD), Air Force, Navy, DARPA, and Northrop Grumman for supporting this work over the years. This book benefited from critique and suggestions made by the reviewers and SPIE staff.

Author's Disclaimer

The views and opinions expressed in this book are solely those of the author in his private capacity and do not represent those of any company, the United States Federal Government, any entity of the U.S. Federal Government, or any private organization. Links to organizations are provided solely as a service to our readers. Links do not constitute an endorsement by any organization or the Federal Government, and none should be inferred. While extensive efforts have been made to verify statements and facts presented in this book, any factual errors or errors of opinion are solely those of the author. No position or endorsement by the U.S. Federal Government, any entity of the Federal Government, or any other organization regarding the validity of any statement of fact presented in this book should be inferred.

Author's Contact Information

Comments on this book are welcome. The author can be contacted at Bruce.Jay.Schachter@gmail.com.

Bruce J. Schachter
March 2020

Chapter 1
Definitions and Performance Measures

1.1 What is Automatic Target Recognition (ATR)?

ATR is often used as an umbrella term for the entire field of military image exploitation. ATR Working Group (ATRWG) workshops cover a wide range of topics, including image quality measurement, geo-registration, target tracking, similarity measures, and progress in various military programs. In a narrower sense, ATR refers to the automatic (unaided) processing of sensor data to locate and classify targets. ATR can refer to a set of algorithms, as well as software and hardware to implement the algorithms. As a hardware-oriented description, ATR stands for automatic target recognition system or automatic target recognizer. ATR can also refer to an operating mode of a sensor or system such as a radar. Several similar terms follow:

AiTR: Aided target recognition. This term emphasizes that a human is in the decision-making loop. The function of the machine is to reduce the workload of the human operator. Most ATR systems can be viewed as AiTR systems in the broader context.

- ATC/R: Aided target cueing and recognition.
- ATD/C: Automatic target detection and classification.
- ATT: Automatic target tracking.
- ISR: Intelligence, surveillance, and reconnaissance.
- NCTR: Non-cooperative target recognition.
- PED: Processing, exploitation, and dissemination.
- SDE: Sensor data exploitation.
- STA: Surveillance and target acquisition.

This chapter sets the stage for the rest of the book. It defines the terms and evaluation criteria critical to ATR design and test. However, every ATR project is different. The terms and criteria presented here will need to be modified to meet the unique circumstances of individual programs. Consider a competition to choose an ATR for a particular military platform. Multiple

Target: Any object of military interest.

Traditional targets are strategic and tactical military craft. This will be the case in point used in this text. However, today, the list can also include improvised explosive devices (IEDs), enemy combatants, human activities, muzzle flashes, fixed sites, commercial vehicles, land minefields, tunnels, undersea mines, and technicals (commercial vehicles modified to contain armament).

Image truth target location or region: A single reference pixel on target or set of pixels on target (target region) as estimated by an image analyst, using ground truth when available.

Bounding box: Rectangle around all of the target or the main body of the target.

For forward-looking imagery, the bounding box is generally rectilinearly oriented (Fig. 1.3). For down-looking imagery, the bounding box will be at an angle with respect to the axes of the image (Fig. 1.4).

For forward-looking imagery, the ground truth target location will generally be pinned to the ground surface rather than at the center of the grayscale mass of the vehicle. This is because the range to the point that the target touches the ground is different from the range along the view-ray through the target center to the ground. This truth will have an associated target location error (TLE) in geographical and pixel coordinates. The TLE for database targets can only be specified statistically. The truthing process might indicate the set of pixels on the target, known as the *target region*. These pixels can match the shape of the target, or be more crudely specified as a rectangular (as in Fig. 1.3) or elliptical region. The target region is generally, but not always, contiguous. A target region can even be smaller than a single pixel, as in the case of low-resolution hyperspectral imagery.

Target report: Report output by ATR generally providing, as a minimum: location in the image of detection (by its reference pixel), the equivalent location as latitude and longitude on an earth map, various categories of classification assigned to the target, and associated probability estimates.

Figure 1.3 Illustrations of a bounding box (a) around the entire target and (b) around the main body of the target.

Figure 1.4 Boxes around targets in overhead imagery can be at any angle.

The information contained in the target report can be quite extensive, but only parts of it can be disseminated due to mission and bandwidth. A popular protocol is MITRE's Cursor-on-Target (CoT). The CoT event data model defines an XML data schema for exchanging time-sensitive positions of moving objects between systems: "what, "when," and "where" information.

Target location and/or region as reported by ATR: Estimated target reference pixel p_{ATR} or region R_{ATR} as provided in an ATR's report.

The ATR will report a target location. This can be the target's geometric center, the center of grayscale mass, the center of the rectangle about the target, the brightest point on the target, or a point where the target touches the ground. The ATR might estimate the pixels on target through a segmentation process. ATR engineers should understand the scoring process and end-user requirements so as to know how best to report a target.

Target detection: Correct association of target location p_{ATR} or target region R_{ATR}, as reported by the ATR, with the corresponding target location p_t or target region R_t in the truth database.

Detection criterion: The rule used to score whether an ATR's reported target location or region sufficiently matches the location or region given in the truth database.

Note that the truth database can contain mitigating circumstances for which the ATR is given a pass if it doesn't detect particular targets. Such circumstances can be: target out of range, not discernable by eye, mostly

1.3 Detection Criteria

It is quite challenging to precisely and unambiguously stipulate what is meant by *target detection*. Let us first consider some relevant terms:

$|R|$ = cardinality of R = the number of pixels in region R.
Let R_t = region on target as indicated by truth data.
R_{ATR} = region on target as reported by the ATR.
p_t = point (or reference pixel) on target according to the truth data, i.e., the target reference pixel.
p_{ATR} = point (or reference pixel) on target as reported by the ATR.
$\|a-b\|$ = distance between points a and b.

First, let us suppose that the ATR outputs a single detection point per object and the truth database contains a single detection point per target. Let

$A = \{p_{ATR}\}$ denote the set of detection points output by the ATR, and
$T = \{p_t\}$ denote the set of detection points in the truth database.

The set C of correct detections output by the ATR is such that each detection in C matches a target in the truth database T according to some match criterion. Here, we will define several common detection criteria, illustrated in Fig. 1.6.

Minimum distance criterion: If the minimum distance between an ATR-reported target point and the nearest target point in the truth database is less than a preselected value d, then the ATR has detected a valid target, as defined by

$$p_t \in C \text{ iff } \min_{p_t \in T} \|p_{ATR} - p_t\| \leq d.$$

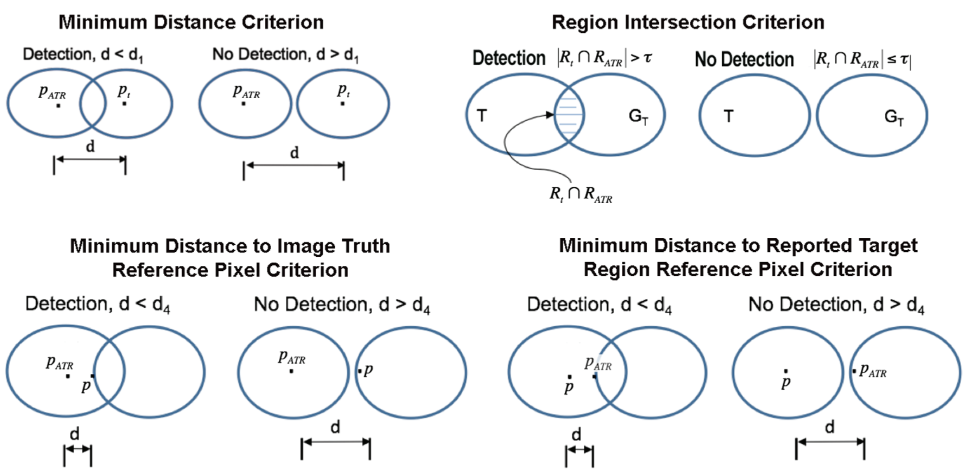

Figure 1.6 Illustration of several detection criteria.

Chapter 2
Target Detection Strategies

2.1 Introduction

An automatic (or aided) target recognizer (ATR) consists of two essential stages: detection and recognition. This chapter covers detection algorithms for literal imagery and ground targets, which are the most basic cases. There are a large number of other cases that aren't covered. Other sensor types (such as vibrometer, high-range-resolution radar, ground-penetrating radar, LADAR, sonar, magnetometer, etc.) and other target types (such as buried landmines, ballistic missiles, aircraft, underground facilities, hidden nuclear material, etc.) require detection algorithms specific to those circumstances. However, the basic strategy covered here still applies.

Several hundred target detection algorithms were evaluated. They were tested on tens of thousands of images of the following types:

- longwave forward-looking infrared (FLIR),
- midwave FLIR,
- Ku-band synthetic aperture radar (SAR), and
- X-band SAR.

Several more-complex algorithms were also designed and tested. Each algorithm is briefly described. The ones that performed best on FLIR imagery are noted. Some insights are shared on target detection with SAR imagery.

Detection approaches with deeper philosophical underpinnings are referred to as *grand paradigms*. Eight grand paradigms are reviewed. In the early days of ATR, there were great debates on the benefits of one paradigm versus another: pattern recognition versus artificial intelligence, model-based versus neural networks, signal processing versus scene analysis, etc. Money flowed to develop the paradigms whose proponents could stimulate the most excitement. Paradigms now generating considerable interest include approaches based on multiscale architectures, biologically inspired designs, and quantum imaging. Each novel paradigm has something to offer. Once the fervor for a new approach dies down, it or its components become additional tools in the ATR system designer's toolbox.

2.1.5 Methodology for algorithm evaluation

The following discussion is based on extensive testing. Test data consisted of target and clutter-blob ROI images, as well as full-sized images. Targets were tracked and wheeled military vehicles. About 30,000 ROIs were used each for LWIR and MWIR tests. ROIs were given approximately constant spatial scaling, taking into account typical range error. The data was from high-end FLIR sensors of the type used on military helicopters and unmanned air vehicles. LWIR imagery was a mixture of data from second-generation scanning sensors and staring sensors. MWIR data was from staring sensors. Depression angles were from 0 deg to 60 deg, with the majority of data from low grazing angles.

Algorithm evaluation was performed in multiple steps. Two equally weighted evaluation criteria were used: parametric and nonparametric.

We recorded each detector's response to target t and clutter c objects. A simplified T-test was used to measure the distance between the two populations of responses:

$$T = \frac{\overline{X}_t - \overline{X}_c}{\sqrt{\sigma_t^2 + \sigma_c^2}}, \tag{2.1}$$

where $\overline{X}_t - \overline{X}_c$ is the difference of the mean scores on targets and clutter, and $\sigma_t^2 + \sigma_c^2$ is the sum of the variances of the scores.

We also examined the tails of the two distributions. The criterion was the percent of clutter blobs that passed through when 90% of targets were detected. Detection algorithms that survived this initial screening were tested on full-frame images. The final evaluation criterion was the list length necessary to detect 90% of targets.

Sufficient data can't be collected to test algorithms against targets and backgrounds in all of their potential diversity. It is more constructive to report on which algorithms tend to perform well than it is to pick out a most excellent algorithm. Algorithms that tested adequately on FLIR data are enclosed in solid boxes in the text.

2.1.5.1 Evaluation criteria for production systems

Algorithm performance is only one of many considerations in algorithm selection for production systems. The software cost of an algorithm is usually determined by the SLOC count required to implement it. Coding and documentation of the code are quite expensive with a coding standard such as SEI Level 5 [or implementation on a field-programmable gate array (FPGA)]. This weighs in favor of simpler algorithms. One must also consider the sustainability of the ATR long after its original developers have retired or moved on to other endeavors. When an ATR enters production, there may not be a contract in place to keep its original design crew intact, analyzing ATR performance year after year, fixing malfunctions, and adapting the statistics of a region near the detector, smaller than the image as a whole, should be used in place of global image statistics. The region should be wider than it is high since background statistics vary with range:

$$T_{10} = \frac{|\overline{X}_1 - \overline{X}_2|}{s_i}, \quad T_{11} = \frac{\overline{X}_1 - \overline{X}_2}{MD_2}, \quad T_{12} = \frac{|\overline{X}_1 - \overline{X}_2|}{MD_2},$$

$$T_{13} = \frac{\overline{X}_1 - \overline{X}_2}{k_2}, \quad \boxed{T_{14} = \frac{\overline{X}_1 - \overline{X}_i}{\sqrt{S_1^2 + S_i^2}}}, \quad T_{15} = \frac{\overline{X}_1 - \overline{X}_2}{s_1},$$

$$T_{16} = \frac{|\overline{X}_1 - \overline{X}_2|}{k_2}, \quad T_{17} = \frac{\overline{X}_1 - \max(\overline{X}_2, \overline{X}_i)}{\sqrt{s_1^2 + \max(s_2^2, s_i^2)}}, \quad T_{18} = \left|\frac{\overline{X}_1 - \max(\overline{X}_2, \overline{X}_i)}{\sqrt{s_1^2 + \max(s_2^2, s_i^2)}}\right|,$$

$$T_{19} = \frac{|\overline{X}_1 - \overline{X}_2|}{s_1}, \quad T_{20} = \frac{(\overline{X}_1 - \overline{X}_i)s_1^2}{\max(s_2^2, s_i^2)}, \quad T_{21} = \left|\frac{(\overline{X}_1 - \overline{X}_i)s_1^2}{\max(s_2^2, s_i^2)}\right|.$$

To reduce computations, simple tests can be designed based only on sample means. In FLIR, these are called tests of temperature difference. Two tests of this type are given, followed by several variations. Note that test T_{22} would not have been a top performer if the test database had a higher percentage of cold targets against warmer backgrounds. Test results must be viewed in relation to operational considerations. Is a cold vehicle a *target*? So, although test T_{23} performed worse than T_{22} on this test set, it is more robust to target polarity and should be considered if operational considerations so warrant. Also note that a 2D difference of Gaussian (DoG) or 2D symmetric Gabor filter can be viewed as a smoothed version of triple-window filter T_{22}:

$$\boxed{T_{22} = \overline{X}_1 - \overline{X}_2}, \quad T_{23} = |\overline{X}_1 - \overline{X}_2|, \quad T_{24} = \frac{(\overline{X}_1 - \overline{X}_2)}{\overline{X}_i}, \quad T_{25} = \frac{|\overline{X}_1 - \overline{X}_2|}{\overline{X}_i},$$

$$T_{26} = \frac{\overline{X}_1 - \overline{X}_2}{\overline{X}_1}, \quad T_{27} = \frac{(\overline{X}_1 - \overline{X}_i)S_1^2}{S_i^2}, \quad T_{28} = \frac{|\overline{X}_1 - \overline{X}_2|}{\overline{X}_1}.$$

To compare the behavior in the central portion of two independent samples of equal size, the difference of the arithmetic means is divided by the inner- and outer-window grayscale ranges R_1 and R_2, respectively. This provides two test statistics similar to the T-test. They are attributed to F. M. Lord (see Ref. 7, pp. 276–277). We also provide four variations:

Chapter 3
Target Classifier Strategies

3.1 Introduction

A target classifier receives image or signal data about a detection point. It infers the category of the object portrayed by the data. The classification decision can benefit from a host of other available information; the more information the better.

ATR often involves a client–contractor relationship. The contractor is committed to providing a quality product to the customer. Yet, target classification is sometimes viewed in a naïve fashion. The customer throws data “over the fence.” The contractor is asked to classify the “targets.” Little thought is given to the breadth and scope of the problem. The usual “solution” involves showing that the contractor’s favorite classifier outperforms several alternatives.

However, the true nature of the target classification problem is more complex. Ironically, choice of a classification paradigm may be the least important aspect of target classification. We will outline the issues involved in target classification. This will be followed by a review of a number of different types of classifiers.

3.1.1 Parables and paradoxes

If no prior assumptions are made about the exact nature of the classification problem, is any reasonable classifier superior to any other? The answer is NO according to the No Free Lunch theorem.[1] Self-deception results from choosing a favorite classifier *a priori* or with limited testing, without a deep understanding of the problem and a well-vetted test plan.

In the absence of encompassing assumptions, is there a best set of features to use for target classification? The answer is NO according to the Ugly Duckling theorem.[2] A good set of features results from understanding the true nature of the problem. Choice of features always biases classifier decisions.

Figure 3.2 Many target types have articulated parts. An ATR must recognize them in all of their variations. The Scud launcher shown here is most dangerous when its missile is in the launch position. (Photo from defense.gov.)

Military vehicles often share components with other similar vehicles or sometimes quite different vehicles. Some vehicle types use the exact same top structure, but the bottom chassis is completely different (e.g., tracked versus wheeled). More commonly, the bottom chassis is the same for a large number of vehicle types, but the top structure is different. In these cases, it is impossible to distinguish the vehicle types when only the common part of the vehicle is observable. Some military aircraft have commercial counterparts.

In the thermal IR part of the spectrum, various parts of a vehicle can appear quite different depending on which parts are turned on or have been recently used, such as engine, exhaust pipe, bogey wheels, driveshaft, internal heaters, or lights.

The critical issue in classifying a target with EO/IR imagery is scale. Without accurate scale information, it is not known whether the target is smaller than a single pixel or larger than the whole image. Is it a hummingbird or a helicopter? What are the sources of scale or, equivalently, range information?

Classifiers must be robust to target variation and varying appearance under different conditions. It needs to be understood that under some conditions and aspect angles, certain vehicle types cannot be discriminated from each other (Fig. 3.3).

3.2.5 Issue 5: Platform issues

The nature of the platform in which the ATR resides affects the target classification problem. Platform vibration is a major issue. Vibration may be dampened by the sensor system, but there is always residual vibration. Vibration is much worse under some circumstances, such as after a missile is

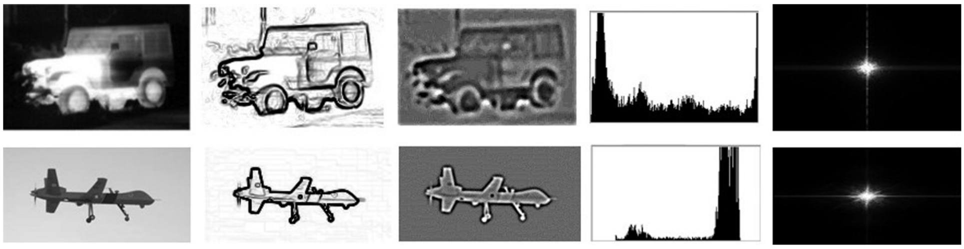

Figure 3.6 Examples of several feature types for an IR image of a jeep and a visible image of a Predator-B UAV. Feature types from left to right are: raw grayscales, edge image, Laplacian image, histogram, and Fourier magnitude. (Jeep image from NVESD.Army.mil. UAV image from Grandforks.af.mil.)

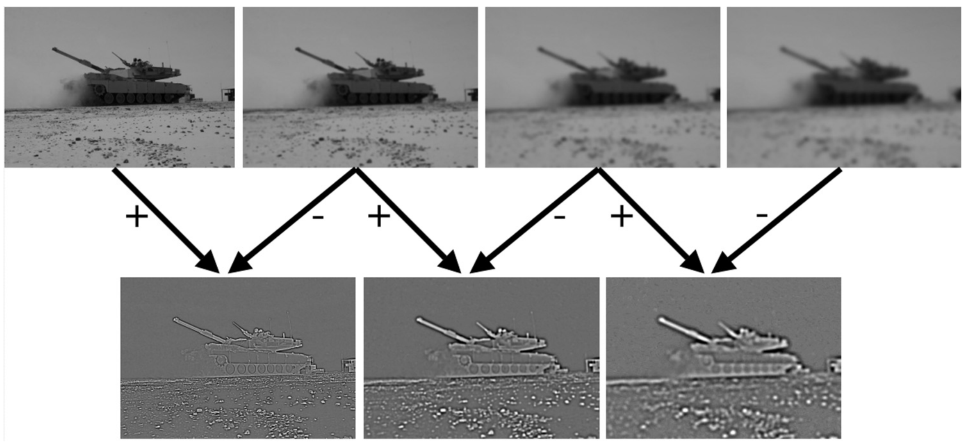

Figure 3.7 Difference-of-Gaussians pyramid. (Tank photo by Sgt. Chad Menegay, www.Army.mil/NewsArchives.)

moments μ_{pq} of order $p+q$ for image region or segmented blob $f(x, y)$ are defined as

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y),$$
$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y); \quad p,q = 0,1,2,\ldots, \tag{3.1}$$

where $\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}$, and $(\bar{x}, \bar{y})$ is the center of grayscale mass of the region.

Features can be formed from combinations of moments.[6] Some of these features are insensitive to translation, rotation, and affine transforms. Other types of moments have been proposed, including ridgelet, Zernike, Gaussian–Hermite, Legendre, Fourier–Mellin, geometrical, and complex.

Features can be obtained by partitioning an image into overlapping blocks and then extracting features local to each block. Popular features of

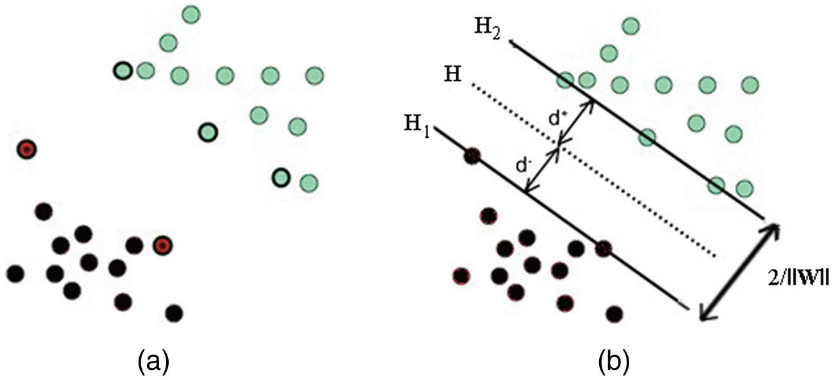

Figure 3.9 (a) Feature vectors from two linearly separable classes. The five support vectors have dark perimeters. (b) The best separating hyperplane, according to SVM, is the one with the most margin, shown here as a dashed line.

$$H_1\colon \mathbf{X}\cdot\mathbf{W}+b=+1,$$
$$H_2\colon \mathbf{X}\cdot\mathbf{W}+b=-1. \tag{3.4}$$

According to the SVM method, the hyperplane H provides the best separation between the two sets of points. The distance between hyperplanes H_1 and H_2 is $2/\|\mathbf{W}\|$. Minimizing $\|\mathbf{W}\|$ maximizes the margin. The following constraint is added:

$$\mathbf{X}_i\cdot\mathbf{W}+b\geq+1\ \text{for}\ \mathbf{X}_i\ \text{of the first class}\ Y_i=+1,$$
$$\mathbf{X}_i\cdot\mathbf{W}+b\leq-1\ \text{for}\ \mathbf{X}_i\ \text{of the second class}\ Y_i=-1. \tag{3.5}$$

The two equations can be combined to yield

$$\mathbf{Y}_i(\mathbf{X}_i\cdot\mathbf{W}+b)\geq1\ \text{for all}\ i,\ \text{where}\ Y_i\in\{-1,\ +1\}. \tag{3.6}$$

This is a constrained optimization problem that can be solved by the Lagrangian multiplier method. The objective is to find the hyperplanes that maximize the margin by minimizing $\|\mathbf{W}\|^2$ such that the discrimination boundary is conformed:

$$\text{Minimize}\ \frac{1}{2\|\mathbf{W}\|^2}\ \text{such that}$$
$$Y_i(\mathbf{X}_i\cdot\mathbf{W})+b=1. \tag{3.7}$$

The problem is formulated in a dual form as

Chapter 5
Multisensor Fusion

5.1 Introduction

Suppose that you are visiting your Aunt Florence. You get hungry and meander into the kitchen. Sitting on the table is a plate of fish. It looks appealing, but something is a bit funky about it. So you poke it, smell it, and taste a tiny bit. It is tasty, but still doesn't seem quite right. Then comes the clincher. Your aunt yells from the next room: "Don't eat the fish!" The human brain is an excellent example of a multisensor fusion system. Fusion of data from your five senses kept you from eating the spoiled fish (Fig. 5.1).

But how did all five senses focus on the same object? This is called the binding problem.[1] All of the features and traits of the fish, in all of the sensor data, must have been segregated from all of the properties of other nearby objects and the background. Then the features must have been associated with the concept of "fish." Binding occurs in many different parts of the brain. There is no single algorithmic solution. Binding is a class of problems: binding over visual space, segregating one sound from others, cross-modal binding

Figure 5.1 All five senses are used to determine whether the fish is too far gone to eat.

classification decision is maximized. The sensor selection problem is often formulated in terms of information theory.

The mutual (shared) information between variables $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2, \ldots, y_o)$ is given by

$$I(X;Y) = \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}, \tag{5.1}$$

where $p(x, y) = p(y, x) = p(x)p(y|x) = p(y)p(x|y)$ is the joint probability density function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y, respectively.

The entropy of random variable X is given by

$$H(x) = \sum_x p(x) \log p(x). \tag{5.2}$$

$H(x)$ can be thought of as the expected information learned from one instance of the random variable X. The mutual information between variables X and Y can be expressed in terms of entropy:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X). \tag{5.3}$$

This is illustrated in the form of a Venn diagram (Fig. 5.14).

The mutual information between class vector $\mathbf{C} = (c_1, \ldots, c_r)$ and feature vector $\mathbf{V} = (v_1, \ldots, v_s)$ can also be expressed in terms of entropy:

$$I(\mathbf{C};\, \mathbf{V}) = \sum_{c_i \in \mathbf{C}} \sum_{v_j \in \mathbf{V}} p(c_i, v_j) \log \frac{p(c_i, v_j)}{p(c_i)p(v_j)} = \sum_{c_i \in \mathbf{C}} \sum_{v_j \in \mathbf{V}} p(c_i, v_j) \log \frac{p(c_i|v_j)}{p(c_i)}. \tag{5.4}$$

Adding a sensor S_2 to a baseline sensor S_1 should improve the classification decision. But that is not the whole story. In order to decide whether it is worthwhile to add the second sensor, one must take into account (1) the redundancy in the features derived from the two sensors and (2) the

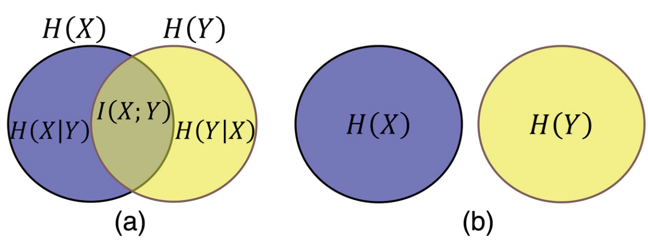

Figure 5.14 (a) The mutual information $I(X;Y)$ of random variables X and Y is a measure of the mutual dependence between the two variables. $H(X)$ and $H(Y)$ are the individual entropies of X and Y, respectively. (b) Two independent variables X and Y have no mutual information.

$$P'(c_k|\mathbf{x}) = \frac{1}{B}[P(c_k)]\prod_{i=1}^{N} P_i(c_k|\mathbf{x}), \tag{5.17}$$

where B is a normalizing constant that is used so that the resulting probabilities sum up to one.

The reason that the optimal Bayes decision rule is seldom used for ATR is that if one classifier outputs a zero probability for a particular class c_k, then it doesn't matter how high a score the other classifiers give it. The product rule is said to suffer from the veto problem in that one classifier can veto the good work of all of the other classifiers.

5.4.2.2 Bayes belief integration

Now suppose that an open test set is available. The N classifiers can be run on the open test set. Each classifier's performance can be reported in the form of a confusion matrix (see Table 5.1), where s_{ij}^k denotes the number of open test set samples from class c_i that were assigned to class c_j by the k^{th} classifier, $k = 1, \ldots, r$. This is done in advance, offline.

Now for online operation, let $\hat{c}^k$ denote the classification decision from the k^{th} classifier. The belief in class c_i is obtained with a product rule,

$$Bel(c_i) = P(c_i)\frac{\prod_{k=1}^{r} P(\hat{c}^k|c_i)}{\prod_{k=1}^{r} P(\hat{c}^k)}, \tag{5.18}$$

where $P(\hat{c}^k|c_i) = P(\hat{c}^k|\mathbf{x} \in c_i)$ is the probability that the k^{th} classifier output is $\hat{c}^k$ given that the unknown $\mathbf{x}$ was really in class c_i. Now using the predetermined confusion matrices s_{ij}^k,[23]

$$P(\hat{c}^k = c_l|c_i) = \frac{s_{il}^k}{\sum_{l=1}^{r} s_{il}^k}, \tag{5.19}$$

$$P(\hat{c}^k = c_l) = \frac{\sum_{i=1}^{r} s_{il}^k}{\sum_{i=1}^{r}\sum_{l=1}^{r} s_{il}^k}. \tag{5.20}$$

Table 5.1 Confusion matrix for the k^{th} classifier.

		Reported by k^{th} classifier		
Truth		c_1	$\ldots$	c_r
	c_1	s_{11}^k	$\ldots$	s_{ir}^k
	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	c_r	s_{r1}^k	$\ldots$	s_{rr}^{jk}

Chapter 8
ATR and Lethal Autonomous Robots

8.1 Introduction

The term *autonomous* stems from the Greek *autos* (self) and *nomos* (law). Autonomous implies reduced human control as compared to automatic (or automated) control.[1] An automated car will be less intelligent and independent than an autonomous car. It will be less able to learn, understand, and adapt to new situations. Driverless and autonomous are nearer to synonyms, as are self-driving and automated. When developed, a truly driverless car will not require a human to take control in difficult situations. It will be able to deliver packages without any humans onboard. Virtually all current automatic (a.k.a. aided) target recognizers (ATRs) leave decisions about weapons delivery to humans. A future autonomous platform could have an embedded ATR, but no humans in direct control. That is the premise of this chapter.

The term *robot* stems from the Czech word *robota* meaning "slave labor." Robots perform labor otherwise done by humans. As we describe them, robots are more mobile than, for instance, robot arms used in manufacturing. Robots are embodied in an engineered structure and situated within the environment. They perform tasks that manipulate the environment and alter their situation within the environment.

A fully autonomous robotic system will change its behavior in response to environmental context and unanticipated events. Doing so will require both dexterity and intellect. An (artificially) intelligent autonomous robotic platform will perform tasks otherwise requiring human intelligence. It will have goals to strive for in the course of its operation. Autonomous robotic machines will be capable of course plotting, navigation, and travel. They will be built from an assemblage of different technologies, including sensors, actuators, motors, computers, and communication devices. Autonomous robots will store energy and use it to initiate and control movement

8.4 LARs and the OODA Loop

The OODA loop is based on the acronym for the cycle of observe, orient, decide, and act (Fig. 8.1). This loop is U.S. Air Force Colonel John Boyd's time-honored model of mental processes.[27] It originally referred to how the brains of fighter pilots are supposed to process information and make decisions. It is analogous to the perception–action cycle known to psychology.[28] The term loop indicates that the process is cyclical. The first three parts of the loop are preparation for correct action. Military advantage is obtained by processing the loop very quickly. The objective is to cycle through the loop faster and more skillfully than one's adversary. This should be possible if you are a robot and your adversary is human.

Consider the elements of the OODA loop as it would be implemented by a LAR (Fig. 8.2).

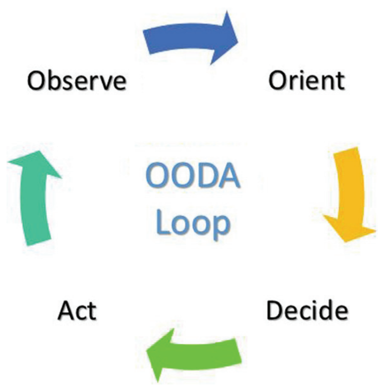

Figure 8.1 The OODA loop is a four-step process: Observe – Orient – Decide – Act.

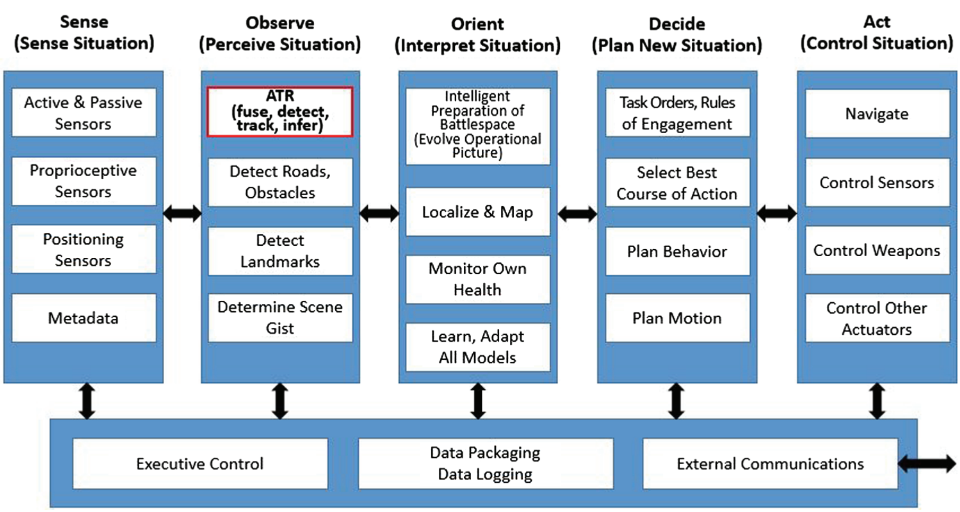

Figure 8.2 A LAR's OODA loop.

class of **x**, by $\text{class}_{\text{True}} = \mathbf{t}^*$. Let t denote a specific incorrect target label, such as "clutter" when the detected object is actually something else like a T-80 tank. The aim of a *targeted adversarial attack* is to find an input $\mathbf{x}'$ such that $f(x') = t$, and for **x** and $\mathbf{x}'$ to appear very similar to each other. The adversary will have been paricularly successful if the difference between **x** and $\mathbf{x}'$ is inperceptible to the human eye. The *untargeted adversarial attack* does not specify any particular target label t. That is, t can be clutter, car, cow, or anything but the correct label.

The attacker's goal is to produce an adversarial example $\mathbf{x}' = \mathbf{x} + \delta\mathbf{x} = \mathbf{x} + \varepsilon$. The general problem of constructing an adversarial example can be formulated as:

Given a clean input **x**,

minimize distortion $D(\varepsilon)$
subject to $f(\mathbf{x} + \varepsilon) = t$; $\mathbf{x} + \varepsilon \in [0, 1]^n$

Distortion $D(\varepsilon)$ is measured by a norm:

$$D(\varepsilon) = \|x - x'\|_p = \left(\sum_{i=1}^{n} |x_i - x_i'|^p \right)^{1/p}.$$

Different norms define different types of attacks.

$p = 0$: The $\mathbf{L}_0$ norm measures the number of mismatched feature elements between **x** and $\mathbf{x}'$.
$p = 1$: The $\mathbf{L}_1$ norm measures the sum of the absolute values of the differences between **x** and $\mathbf{x}'$.
$p = 2$: The $\mathbf{L}_2$ norm measures the Euclidean distance between **x** and $\mathbf{x}'$.
$p = \infty$: The $\mathbf{L}_\infty$ norm measures the maximum difference between x_i and x_i' $\forall\, i$.

One possible goal of the attacker is to develop adversarial examples that minimize $D(\varepsilon)$, per the chosen norm, to make the adversarial examples undetectable to human vision, but still to fool a classifier. Several approaches have been proposed to find adversarial examples. These approaches are often referred to by their acronyms: BFGS, Deep Fool, FGSM, BIM, JSMA, Carlini–Wagner, etc.

8.8.8 Ways to mitigate adversarial attack against a LAR (particularly its ATR component)

1. Develop the ATR in a secure environment with cleared personnel.
2. Do not let anyone (without clearance and need to know) borrow or even see the training data, test data, data collection plan, or T&E documents. Do not let anyone (without clearance and need to know)