Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Development of an ATR Workbench for SAR Imagery

Implementing a single user interface for automating target recognition processes

R.A. English, S.J. Rawlinson and N.M. Sandirasegaram

Defence R&D Canada - Ottawa
TECHNICAL REPORT
DRDC Ottawa TR 2002-155
December 2002

Canada

Development of an ATR Workbench for SAR Imagery

Implementing a single user interface for automating target recognition processes

R.A. English
Defence R&D Canada – Ottawa

S.J. Rawlinson
MacDonald, Dettwiler and Associates Ltd.

N.M. Sandirasegaram
Defence R&D Canada – Ottawa

Defence R&D Canada – Ottawa

Technical Report

DRDC Ottawa TR 2002-155

December 2002

Abstract

In order to assist in the development of automation techniques for target recognition within SAR imagery, we have implemented an evaluation platform, the ATR Workbench. This will allow researchers and Geospatial Image Analysts (GIAs) to investigate the capabilities of various commercial and experimental applications, singly or in combination, as applied to the target recognition process. The platform will enable studies to determine which aspects improve GIA performance when automated, which methods best improve classifier performance, as well as which methods work better for particular environments and target class definitions.

The ATR Workbench was developed, using several open source tools, to provide a platform independent bridge between Automatic Target Detection (ATD) applications and target classifiers. It is capable of importing several kinds of ATD reports, of applying different feature extraction and preprocessing algorithms and of implementing various aspects of Automatic Target Recognition (ATR) applications while importing, displaying and reporting their results. Each step may be automated or operated interactively, as required. Initially, this capability is demonstrated on imagery based upon the public MSTAR data set.

Résumé

Afin d'aider au développement de techniques d'automatisation pour la reconnaissance des cibles à l'aide de l'imagerie RSO, nous avons mis au point une plate-forme d'évaluation appelée *ATR Workbench.* Cet outil permettra aux chercheurs et aux analystes d'images géospatiales (AIG) d'étudier les possibilités offertes par diverses applications commerciales et expérimentales, de manière individuelle ou regroupée, pour le processus de reconnaissance des cibles. Cette plate-forme permettra d'effectuer des études afin de déterminer quels aspects, quand ils sont automatisés, améliorent le travail des AIG, quelles méthodes améliorent la performance des algorithmes de classification, et quelles méthodes conviennent le mieux selon l'environnement et les définitions des classes de cibles.

L'*ATR Workbench* a été mis au point à l'aide de plusieurs outils du domaine public, afin d'en faire une passerelle, indépendante des plates-formes informatiques, entre les application de détection automatique des cibles (DAC) et les algorithmes de classification des cibles. Il peut importer plusieurs types de rapports DAC, appliquer différents paramètres d'extraction d'entités et algorithmes de prétraitement et appliquer divers aspects des applications de reconnaissance automatiques des cibles (RAC) tout en important, affichant et rapportant leurs résultats. Chaque étape peut être automatisée ou exécutée de manière interactive, au besoin. Au début, ces possibilités seront démontrées avec une imagerie basée sur l'ensemble de données MSTAR, qui sont du domaine public.

This page intentionally left blank.

Executive summary

Currently, analysis of radar imagery is primarily achieved through the efforts of human geospatial and image analysts. Synthetic Aperture Radar (SAR) imagery is a projection of range data onto a plane parallel to the sensor line of sight, rather than perpendicular as occurs with human vision, photography and EO/IR imagery. Extensive training, experience and effort are therefore already required to achieve good analysis of SAR imagery. For this reason, among others, computer automation of some or all of the analysis is becoming increasingly desired to allow operational personnel to concentrate their efforts elsewhere, so long as the performance and efficiency of the analysis can be maintained or improved.

The core technologies underlying the automation of target recognition, such as pattern recognition, machine vision, artificial intelligence, etc., are still considered to be in their infancy and are subject to active research on fundamental principles. As a result, no theory of Automatic Target Recognition (ATR) has yet been successfully derived, and is unlikely to be until the foundations of the enabling technologies are themselves well understood. The lack of a theory is problematic for implementing any ATR application, because without one there can be no quantitative prediction of performance. However, qualitative prediction may be achieved by building up empirical evidence and experience with a given application.

To this end, MacDonald Dettwiler and Associates worked under contract with DRDC Ottawa to develop a software tool linking SAR target classifiers with target detectors, enabling interactive or automated application of each step of the target recognition process. This so-called ATR Workbench integrates various ATR software with intermediate processing algorithms through a single user interface that will allow target selection, orientation, segmentation and classification to be implemented by the user or by one or more automation module. The ATR Workbench interfaces with external applications, including the OMW ship detector, the HNeT classification engine and various Matlab-based components.

As an R&D tool, the ATR Workbench is in a state that will allow: a) the impact of automation on target recognition methodologies currently used by the Canadian Forces (CF) to be examined, b) the development of standardized empirical evaluation techniques for SAR ATR applications in terms of effectiveness on CF capabilities, c) effectiveness evaluation of specific SAR ATR algorithms, d) better exposure of operational personnel to the capabilities of various automation processes as part of demonstrations of end-to-end ATR systems, e) a baseline for the operational community to provide feedback on further research, development and implementation issues concerning SAR imagery analysis and exploitation, and f) opportunities for operational personnel to develop confidence in the application of specific SAR ATR algorithms to specific operational tasks, amongst others. Further development of the software is expected to allow the ATR Workbench to be configured to provide the prototype for any number of future ATR systems that could be implemented by the CF to enable automation for operational target recognition requirements.

R.A. English, S.J. Rawlinson, N.M. Sandirasegaram. 2002. Development of an ATR Workbench for SAR Imagery. DRDC Ottawa TR 2002-155. Defence R&D Canada – Ottawa.

Sommaire

À l'heure actuelle, l'analyse des imageries radar est essentiellement le fait de personnes spécialistes en images et en caractéristiques géospatiales. Les images prises par radar à synthèse d'ouverture (RSO) consistent en une projection de données en distance sur un plan parallèle à la ligne de visée du capteur, plutôt qu'à la perpendiculaire comme c'est le cas avec la vision humaine, la photographie ou l'imagerie EO/IR. Pour bien analyser les images RSO, une formation poussée, de l'expérience et un certain effort sont requis. C'est pourquoi, entre autres, l'automatisation informatisée totale ou partielle de l'analyse est hautement souhaitable, afin de permettre au personnel opérationnel de concentrer ses efforts ailleurs, en autant que l'on puisse maintenir ou améliorer le rendement et l'efficience de l'analyse.

Les technologies de base sous-jacentes à l'automatisation de la reconnaissance des cibles, comme la reconnaissance des formes, la vision machine, l'intelligence artificielle, etc., en sont encore à leurs premiers balbutiements et font l'objet de recherches fondamentales. Par conséquent, on ne dispose pas encore d'une théorie bien établie de la reconnaissance automatique des cibles (RAC), et il est probable que ce sera le cas tant que les fondements mêmes de ces technologies ne seront pas eux-mêmes bien compris. L'absence d'une telle théorie pose donc un problème pour l'implémentation de toute application RAC, car il est impossible de faire des prévisions quantitatives pour l'efficacité d'une telle application. Toutefois, il est possible de faire des prévisions qualitatives en acquérant des preuves empiriques et de l'expérience avec une application donnée.

À cette fin, MacDonald Dettwiler and Associates ont mis au point, dans le cadre d'un contrat avec RDDC Ottawa, un outil logiciel qui fait le pont entre les algorithmes de classification de cibles sur les images RSO et les détecteurs de cibles, permettant ainsi l'exécution interactive ou automatique de chaque étape du processus de reconnaissance des formes. Cet outil, appelé *ATR Workbench* intègre divers logiciels de RAC et des algorithmes de traitement intermédiaire, grâce à une seule interface utilisateur permettant à celui-ci ou à un ou plusieurs modules d'automatisation de sélectionner, orienter, segmenter et classifier les cibles. L'*ATR Workbench* assure une interface avec des applications externes, dont le détecteur de navires OMW, le moteur de classification HNeT et diverses composantes basées sur Matlab.

Outil de R et D, le *ATR Workbench* permettra, dans son état actuel, les tâches suivantes : a) étudier l'impact de l'automatisation sur les méthodes de reconnaissance des formes actuellement utilisée par les Forces canadiennes (FC) ; b) mettre au point des techniques d'évaluation empirique normalisées pour les applications RAC avec images RSO, en termes de leur efficacité pour les capacités des FC ; c) évaluer l'efficacité des algorithmes RAC spécifiques pour les images RSO ; d) mieux faire connaître au personnel opérationnel les possibilités des divers processus d'automatisation dans le cadre de démonstrations de systèmes RAC entièrement automatisés ; e) offrir un système de référence au personnel opérationnel pour qu'il puisse faire connaître ses observations et commentaires sur les orientations futures touchant la recherche, le développement et l'implémentation des systèmes d'ana-

lyse et d'exploitation de l'imagerie RSO ; f) offrir au personnel opérationnel la possibilité d'acquérir une certaine confiance dans l'application d'algorithmes RAC RSO spécifiques.

On prévoit que le développement du logiciel permettra de configurer l'ATR Workbench afin d'en faire un prototype pour tout nombre de futurs systèmes RAC que les FC pourraient implémenter, compte tenu de ses exigences en reconnaissance opérationnelle des cibles.

R.A. English, S.J. Rawlinson, N.M. Sandirasegaram. 2002. Development of an ATR Workbench for SAR Imagery. DRDC Ottawa TR 2002-155. R & D pour la défense Canada – Ottawa.

This page intentionally left blank.

Table of contents

List of figures

List of tables

Acknowledgements

The authors would like to thank Dr. Jeff Secker for his support in providing descriptions of his upcoming FuseBox: Land ATD software, for making available a simulated Fuse-Box detection list generated on the demonstration image used herein and his many useful comments and observations.

The efforts of the Open Source software community are also appreciated, especially those providing support for OpenEV, Python, PyGTK and GTK.

DARPA, the COMPASE Center and AFRL have made publically available significant parts of their MSTAR data set along with their standard evaluation method, which has become a invaluable tool for the ATR R&D community.

This page intentionally left blank.

1 Introduction

Creating intelligent systems capable of mimicking traits found in humans is a challenge that researches have been working on for decades [1, 2]. There has been some limited success in applying artificially intelligent systems to specific domains, and to recognition problems in particular. For example, optical character recognition [3, 4], speech recognition [5, 6] and, more recently, in face recognition [7, 8]. The field of automatic target recognition (ATR) has many new challenges, but it is hoped that some of the knowledge gained from these related domains can be successfully applied to ATR. The goal of this project has been to develop tools for ATR research in Synthetic Aperture Radar (SAR) and to create a baseline application that Defence R&D Canada (DRDC) clients and associates can discuss and provide feedback on. It strives to be both a platform to promote dialog and a platform to facilitate research.

This document will explain the problem from both the mathematical and technological perspective. It will then describe the methods used to implement a solution, including the technology architecture and implementation details. Finally, some results and the current capabilities of the software tool will be discussed.

1.1 Background

Currently, analysis of radar imagery is primarily achieved through the efforts of human Geospatial and Image Analysts (GIAs). Unlike photography, Electro-Optical (EO) or Infrared (IR) sensors, which generate imagery on a plane perpendicular to the Line of Sight (LoS) as occurs in human vision, SAR imagery is a projection onto a plane that is coincident with the LoS [9] as shown in Figure 1. Thus, interpretation of SAR imagery is not intuitive for human analysts and will require significant concentration and care to properly analyze the imagery. Extensive training, experience and effort are therefore already required to achieve good analysis of SAR imagery. For this reason, among others, computer automation of some or all of the analysis is becoming increasingly desired in order to allow personnel to concentrate their efforts elsewhere, so long as the performance and efficiency of the analysis can be maintained or improved [10].

Many attempts at providing ATR for imagery have not been able to maintain performance criteria over sufficiently general conditions [11]. The core technologies underlying the automation of recognizing targets in imagery, such as pattern recognition, machine vision, artificial intelligence, etc., are still considered mostly in their infancy. Often the science they are built upon is itself not well enough understood and subject to active research on fundamental principles [12]. As a result, no theory of ATR has yet been successfully derived, and is unlikely to be until the foundations of the enabling technologies are themselves successfully understood [13].

The lack of an ATR theory is problematic for implementing any ATR application because without a theory no prediction of performance can be derived. In other words, for a given performance requirement, one cannot quantitatively determine in advance whether an appli-

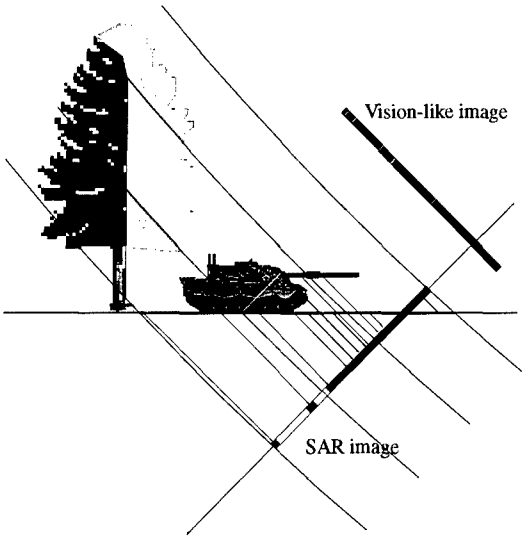

Figure 1: *Geometry for remote sensing a tank. Vision-like imagery is perpendicular to the line of sight, giving an unobscured view of the tank (green bar). SAR imagery is a projection onto the LoS, thereby mixing ground, tank and tree information.*

cation will be able to do the job or not. Often, a given ATR application will perform suitably well for a specific test, but then inexplicably fail under operational conditions different from the test conditions [14]. The only way to know for certain whether an application will work under any given set of conditions is to apply it to those conditions. However, a qualitative prediction of an application's performance may be achieved by building up empirical evidence and GIA experience with it. Furthermore, it is envisioned that this qualitative assessment will become a necessary part of the scientific development of the ATR applications themselves, providing a measure of improvement [15], at least until an ATR theory becomes available.

To achieve this end, it is necessary to establish a testbed for the automation processes to be applied as a whole or in part in such a way that GIAs can obtain the required empirical results and experience with a variety of ATR applications and subprocesses. This so-called ATR Workbench is being developed towards achieving this goal by providing access to multiple ATR methods via a single graphic user interface (GUI) implementation, in order to classify targets originally detected manually within the ATR Workbench or as output from an external ATD application. Initially, two ATR methods previously developed by DRDC Ottawa are supported, along with interactive classification. As such, the ATR Workbench can be considered to address 5 of the 35 requirements initially set for the much more ambitious Operational Digital Imagery Navigator (ODIN) project [16], but restricted to SAR imagery. ODIN aimed to interface a multitude of Commercial Off-the-Shelf (COTS) analysis software for multiple sensors, data fusion and ATR to assist operational image analysis for the Canadian Navy.

The starting point for the ATR Workbench is a processed SAR image scene and possibly the output from a pre-existing target detection program, which provides a list of target locations within the image. DRDC Ottawa has been investigating several ATD applications, some of which the ATR Workbench can use as ATD modules by supporting the import of the applications' report files.

The most mature of these is the Ocean Monitoring Workstation (OMW), which was developed in collaboration with Satlantic who is making the product commercially available [17, 18]. The OMW is a detector for ships on the open ocean imaged by satellite SAR sensors.

Another target detection application is under development by DRDC Ottawa as part of an image fusion workstation [19]. Phenomenological segmentation of a scene is achieved and the location of target segments reported. Currently, the fusion workstation operates under MATLAB, also allowing an easy interface for the ATR Workbench.

At the other end of the system, DRDC Ottawa has designed classifiers for ground targets [20, 21] that have been shown to successfully discriminate between the target classes defined by the standard evaluation criteria published by the United States Air Force Research Laboratory (AFRL) to be used with the publicly released set of Moving and Stationary Target Acquisition and Recognition (MSTAR) images [22].

Using AND Corporation's Holographic Neural Technology (HNeT) [23] as the classifier engine, discrimination between target classes is achieved by determining invariant Fourier coefficients from the magnitude imagery of the training vehicle. Other feature types, such as Zernike moments and wavelet coefficients are under investigation, but have yet to attain as successful results as the Fourier coefficients. Meanwhile, a Multi-Layer Perceptron (MLP) Neural Network (NN) classifier has been developed by DRDC Ottawa to again discriminate target classes using invariant Fourier coefficients [21].

A ship classifier using the HNeT engine and exploring Haar, wavelet and Zernike moment invariants is under development using ship data collect by the DREO XDM SpotSAR radar [24]. This classifier will be incorporated into the ATR Workbench at a later date.

Thus, the challenge for constructing the ATR Workbench has been to implement appropriate automated and manual techniques to preprocess parts of the imagery scene at each detection location, so that the target can be evaluated by the user or sent to one or more classifier modules. Compilation, display and reporting of the classification results then provides the user the necessary information to qualitatively assess performance of the ATR applications.

1.2 Scope of the Problem

1.2.1 Criteria for ATR in SAR imagery

Fundamentally, the requirement for automating target recognition is to determine the appropriate characteristic function, χ, for each target class, C, so that for any target object, t,

we have

$$\chi_C(t) = \begin{cases} 1, & t \in C, \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

In the case of ATR on SAR imagery, we are removed from the above ideal case in that we do not have access to the actual target object, but only to a limited number of images, $\{I_n^t\}_{n=1}^N$, each containing a representation of the object. Each image representation contains only a subset of information about t, and that information is often merged with or modified by information particular to the object's surroundings (clutter) and the sensor (noise) at the time of imaging.

In an attempt to re-establish separation of the target information from all other sources, feature extraction is performed, i.e., specific properties $\{f_m\}_{m=1}^M$ are calculated, according to their expected dependency on t alone,

$$f_m\left(I_{n_1}^t\right) = f_m\left(I_{n_2}^t\right), \qquad \text{for all possible } n_1, n_2. \tag{1.2}$$

Thus, actual image ATR applications operate using a projection of the original characteristic function onto the subspace of information spanned by the selected feature set. Hence, instead of (1.1), we need to obtain a function

$$\hat{\chi}_C\left(f_1(I_n^t), f_2(I_n^t), \ldots, f_m(I_n^t), \ldots\right) = \begin{cases} 1, & t \in C, \\ 0, & \text{otherwise,} \end{cases} \tag{1.3}$$

valid for every image I_n^t. Unfortunately, it may be that (1.3) does not have a solution. There may exist, for example, $t_1 \in C$ and $t_2 \notin C$ such that $I_{n_1}^{t_1} = I_{n_2}^{t_2}$, which would yield identical arguments on the left hand side of (1.3) but require opposite solutions on the right.

Furthermore, the images are also characterized by the physical geometry involved between object and sensor for each data acquisition as well as the image processing techniques used to generate each I_n^t. Since we are using only a finite number of images, it is not possible to adequately account for the results of $n \notin \{1, \ldots, N\}$. Therefore, in order that (1.2) be satisfied, it is necessary to ensure that each source of additional information is, as much as possible, either 1) *independent* of both n and t (e.g., use of the same image processing techniques for all images), or 2) *variation* as a function of n identically for each t (e.g., always collect imagery for each target from the same specified geometries).

In practice, we use only 1) a finite subset of C, $\{t_1, \ldots, t_Q\} \subset C$, namely those objects for which the appropriate imagery exists, 2) a finite subset $\{I_n^{t_q}\}_{n=1}^{N_q}$ for each t_q, viz., a finite number of images for each of those objects, and 3) $\{f_m\}_{m=1}^M$, a finite subset of features to be extracted. While the first two choices are due to limitations on the availability of data, the third choice is made due to computational limits on time, memory and storage, as well as issues of complexity, independence and derivation of the features [25]. It is, therefore, the determination of an appropriate set $\{f_m\}_{m=1}^M$ that is at the heart of selecting one ATR system over another. The choices are varied and without much theoretical guidance.

Most ATR systems require feature sets to be preselected manually and are chosen based on empirical evidence for being both able to discriminate classes and invariant to clutter and

noise [26, 27]. Such a selection method may aim to mimic what a human analyst will look for in making a decision, or may be based on an analytical investigation of the problem itself. Another method is to select a space of potentially discriminating feature and then use the training data to determine invariance within the feature space. The most invariant features are then selected as the feature set. It is the latter method which we pursue for the initial ground-based ATR classifiers.

1.2.2 Criteria for ATR Workbench Software

In order to achieve the ATR criteria given in Section 1.2.1 under operational conditions encountered by GIAs, researchers require a method by which to examine the effects automation applications have on the current target recognition process, without hindering the current capabilities of the GIAs. To this end, the concept of the ATR Workbench was derived.

A number of requirements have been developed for the ATR Workbench software to support this research (see Annex B). Specifically, the software must be modular, integrate with existing technologies, be based on established libraries, be quickly developed and run under common computer operating systems.

Modular systems are well suited for experimentation tasks because they permit easy replacement of system components for evaluation. Ideally, components could be swapped into and out of the system framework for testing without affecting other parts of the system. This requirement also implies that the interfaces between modules needs to be well defined and easy to use. The format for the exchange of data needs to be simple to read and write.

A number of ATR applications (HNeT, OMW) previously developed are the basis for particular modules. Therefore, the system required easy *integration with existing technologies* as well as with any future applications developed. To a certain extent the modularity of the system facilitates this, but the choice of programming language used to implement the modular is also an important consideration. In essence, the language needs to "glue" the separate ATR application modules into a cohesive system.

To avoid duplicating previous efforts and to focus resources on the ATR problem, the application was built upon *established software libraries* that provide a solid foundation. These libraries provide the low-level foundation upon which the ATR Workbench was built. Access to the library source code is an advantage for customization (if necessary) and fixing bugs. Documentation and support for the libraries also factor for this requirement. Furthermore, the libraries need to be capable of supporting vector and raster graphics, and need to support many of the standard geographic file formats for compatibility.

The application must be *developed rapidly* and easily modified as future requirements dictate. Application speed is not an important consideration, but development time and functionality are. Therefore a scripting language such as Python and Matlab were considered.

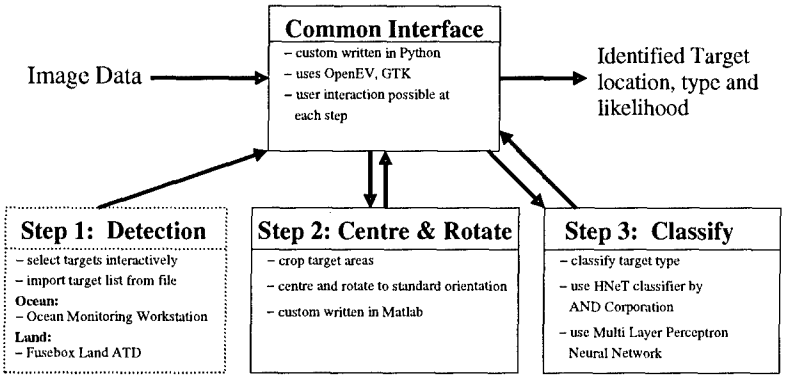

***Figure 2:** Module System Design*

Since it is unclear which operating system the application will be required to run under, the software *supports multiple-platforms* (i.e., UNIX and Windows). This requirement has a significant impact on the choice of programming language, libraries and, in particular, the GUI (Annex B).

2 Method

2.1 Software Design

The chosen system design (Annex A) consists of four modules: a detector, pre-processor, classifier and the GUI. Each module corresponds to a step in the processing chain, except for the GUI and its underlying framework that connects all the steps. The processing chain can be completely controlled from the GUI. This permits an interactive environment where an image analyst can evaluate the results from each step, correct them if necessary, and then re-initiate the processing.

Figure 2 provides a high level view of the modules and their relationships. First, the image data is read in by the GUI and displayed for the user. Next, if desired, a detection module is initiated that imports potential targets from a pre-existing detection report on the image and highlights the location on the image in the GUI. Otherwise, the user may interactively select target locations. The user is able to examine the results of the detection step and make any desired alterations before starting the next step in the chain. Using the GUI, detected targets are extracted one-by-one as subimages and subjected to the image pre-processor that performs analysis and re-formatting of the subimage before it is sent to the classifier. The classifier is responsible for determining measures of likelihood that the target belongs to pre-set target classes. These results are then returned to the GUI for display to the user.

The open-source programming language, Python was chosen to write the GUI and ATR Workbench framework. Although not widely known, Python is a general purpose scripting language, similar in many ways to Perl or MATLAB [28]. One of the primary reasons

for choosing Python was because of the geographic library, OpenEV, and the GUI library, PyGTK, both of which are supported on a number of different platforms (see Section 2.2). As a scripting language Python also provides a development environment that facilities rapid proto-typing of algorithms and easy modification. Its strengths are integration and support through well developed libraries, rather than speed that a compiled language like C would provide. As well, Python is supported on most operating systems currently available to the public.

To integrate existing software, Python has the capability to make operating system level calls to start applications. Typically, the ATR Workbench specifies input file locations via the command line to applications when they are started by Python. Output from the programs is written using a known filename that is then ingested by the Python GUI.

In the case of HNeT under Windows, the native COM/ActiveX automation interface is used. This provides Python with direct access to the many functions of HNeT and allows for greater control of the classification process. Although HNeT is not currently available for UNIX platforms, a Linux version is planned and a Python wrapper could be used to achieve a similar interface, or HNeT could be implemented using operating system level calls as above.

2.2 GUI

The most difficult challenge in implementing the GUI is that the application has to be supported on both Windows and UNIX platforms. Typical windowing libraries, such as Microsoft's MFC and X-Window's xlib, are very dependent on their respective operating systems. This challenge was recognized early on in the design process based on previous project experiences. Therefore, selecting an appropriate combination of programming language, graphics and window library that were easily portable across operating systems was a primary consideration.

Several options were considered for the GUI (Annex B) such as MATLAB, IDL Java and Python. The final evaluation revealed that the combination of Python with OpenEV and PyGTK best met all the requirements. They are proven to work on both UNIX and Windows systems, OpenEV has extensive support for geographic raster and vector formats, supports large images and the Python language makes for quick development. GTK was a natural choice as it was already being used by OpenEV, has been integrated for use in Python and provided all the basic window library features required.

Figure 3 shows the technology dependencies of the application GUI and how they are all related. The ATR Workbench code is written in Python and depends on two main libraries, PyGTK and OpenEV. OpenEV itself also depends on PyGTK to a certain extent to provide services and depends on OpenGL for graphics. The lower levels are in C, with the upper level that the ATR Workbench depends on in Python.

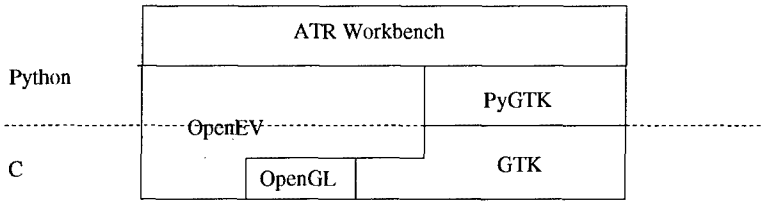

***Figure 3:** ATR Workbench GUI Dependencies*

2.2.1 PyGTK

The PyGTK library provides all the GUI objects a user typically sees and interacts with for an application. For example, it provides a window object that can contain other objects such as a menu bar, buttons, an image display area, text box etc. The library also provides an event handling mechanism whereby a function is called when a user takes particular actions. For example, when a mouse click occurs over a button the associated function is executed. PyGTK is essentially just a Python interface to GTK. Furthermore, it is an open source library, freely available for the development of and use by applications based upon it.

2.2.2 OpenEV

OpenEV was developed by Atlantis Scientific Inc. The name OpenEV is used for both the geographic image library and a reference application that demonstrates the library's capabilities. Any references to OpenEV in this document are to the library only.

The OpenEV library provides two types of image layers, raster and vector. It supports reading many types of geographic raster formats and it supports both reading and writing for vector layers [29]. Tools to draw and edit vector layers are available through the library, as are tools to zoom and pan the view area. The view area can display images with their associated geographic projection. The PyGTK GUI provides the user with control over the operation of OpenEV's functions in support of each step of the ATR process.

Since OpenEV is an open source initiative, it is freely available for both the development and use of applications based upon it [30, 31].

2.2.3 Layout and Controls

The interface of computer applications has a large impact on how usable the tool is to human operators, and ultimately how successful the tool is at accomplishing its task [32]. For well-established tools the interface has usually evolved over time and a standard developed. This facilitates a user's ability to quickly learn a new or slightly different tool, as the interface is already familiar. The automobile is an excellent example of this. Although the location of pedals and controls varied considerably when cars were first invented [33], the controls standardized over time with user feedback. Today's standard automobile controls make it

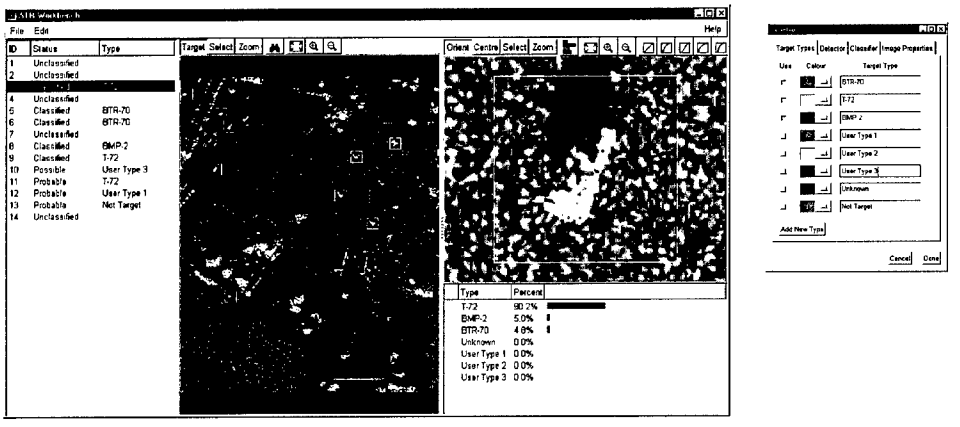

Figure 4: *Screen capture of ATR Workbench showing the GUI layout (left) and Setup menu (right). A detailed example of the ATR Workbench operation is given is Section 4.*

relatively easy to drive cars from different manufacturers, or to drive a new model. Just like automobiles in the past, computers interfaces are becoming increasingly standardized [34]. New software applications are therefore advised to follow existing standards or paradigms were applicable to ensure maximum usability [35].

In the absence of a well-developed standard interface for ATR, a reasonable design approach would be to follow the logical steps for the task that the human GIA would have done, thereby allowing the operators to relate their real world experience with the process to that of the automated application. By providing the user the ability to implement each step interactively, the initial ATR Workbench user interface provides a basis for which user feedback can be generated to guide development of the application accordingly.

To illustrate this process, a screen capture of the GUI is presented in Figure 4. It shows the chain of steps loosely displayed in a left to right fashion. High-level information for the scene is on the left, with more details as you move right. In the far-left panel is a list of potential targets, their current status (unclassified, classified [automated], possible, probable, unknown) and a target class type if it has been classified. This panel is updated as new information is generated or input by the user. In addition to the class types predetermined for the classifier training definitions, new class types may be made available by defining by them under the Setup menu. Two special pseudo-class types are also available (see Annex C). The "Not Target" type allows the user to indicate that a detection is likely a false alarm, while keeping the location as part of the detection list and any reports to be generated. The "Unknown" type may be selected by the user to indicate that the appropriate class for the target cannot be determined or may assigned by a classifier to indicate that the target likely belongs to none of the pre-determined types. Selecting an item in this panel's list also displays the item's detailed information and view in the following windows.

The second pane is the image overview window. It displays the image being analyzed and the locations of potential targets. If a potential target has been classified, the marker (an

open box) for its location is highlighted in a particular colour. The user assigns each target class a colour. The colour coded targets in the overview window give the user a very quick and easy way to interpret the relative locations of targets in the scene and which targets remain unclassified. Tools above the window give the GIA the ability to zoom in and out, select particular targets, and manually include new target locations in the scene that may have been missed by the detection process.

The next panel is the zoom window at the top right. Its purpose is to display a single target, compared to the previous window that gave an overview of many targets. This close up view allows the user to better manipulate the image, to manually classify the target or to verify the correctness of the automated classifier results. It also serves as an input window for the manual rotation and centering of the target prior to sending the image to one of the automated classifiers.

In the bottom right is a panel that displays the classifier output. Unfortunately, the limitations of the width of the screen prevented this panel from being placed to the right of the zoom window, to preserve the left to right increase in detail. Currently, a table format is used, with three columns: class types, numerical likelihood measures and horizontal bar graphs. The bar graphs provide a visual representation of the relative measures and make for easy comparisons by a GIA. The class types displayed in the panel are all the types that have been defined in the ATR Workbench Setup. The automated classifiers will generate zero likelihoods for class types that have not been pre-determined and trained upon, except for the Unknown type, for which a likelihood measure of the target belonging to none of the pre-determined types is calculated as described in (2.12).

Information about the classification results from an ATR module is also incorporated in other displayed aspects. The colour outlines in the zoom and overview windows are updated to reflect the fact that the target has been classified as a particular type. Additionally, the list in the leftmost panel is updated with the new target type.

The user has the option to manually override the type assigned by the classifier. A dialog box for manual intervention is available by right clicking on a target item from the list in the leftmost panel. When a GIA classifies the target, the status of "Possible," "Probable" or "Unknown" is used to indicate the level of confidence. The term "Classified" is reserved to indicate a decision from an ATR module.

The initial design also called for two additional zoom windows so that multiple targets could be displayed simultaneously for comparison and parallel analysis. They not been included in the initial version due to the additional complexity of many windows, but will hopefully be implemented in a future version.

In general, colour and visual representations have been used where possible as this allows the user to more quickly and easily interpret information from the application. Optimizing these visual representations is an area of active research, towards which the ATR Workbench may contribute. Each panel's size can be changed by dragging the dividers, allowing the user determine the appropriate size of each window.

Since the panels, from left to right, roughly follow the steps involved in target recognition, they should provide a natural way to view the process. For each of these steps information is fed back to the previous panels once new information about a target is available.

Good user interfaces are challenging to design, particularly when no established models exist. It is anticipated that an iterative process involving feedback from the GIAs will help to refine the interface to make it as operationally efficient and friendly as possible.

2.3 Detection

Initially, several basic formats of target detection lists are considered for importation into the ATR Workbench, corresponding to current ATD applications being studied.

The detection step is primarily concerned with identifying locations in the image that are possible targets. Subimages at these locations will later be sent to the classifier to attempt to determine the type of target. Through the GUI the operator can view the output from the detection step and remove any obviously false targets or make any desired additions.

2.3.1 Ocean Monitoring Workstation

The OMW (Figure 5) was developed by Satlantic in collaboration with DRDC Ottawa and is a currently available as a commercial product. Currently, the OMW requires imagery to be input in CEOS format and supports RADARSAT 1 and ERS 1/2 data. It uses a k-distribution model with a Constant False Alarm Rate (CFAR) method to detect potential targets, while size and shape are used to eliminate some false alarms [36]. When land is present in a scene, a geolocated land mask is generated to restrict the detector to ocean pixels. The basic technology involved also uses Radon Transforms to detect wake signatures. From a ship's wake, speed and direction of the target can often be determined, although the ATR Workbench does not currently incorporate this information. The detected locations determined by the OMW are then output to a report file that the ATR Workbench reads.

2.3.2 FuseBox: Land ATD

FuseBox is an image fusion system being developed at DRDC Ottawa, and the Land ATD toolbox includes algorithms for the ATD of stationary land targets. It is the first toolbox to be implemented, with the other planned toolboxes in FuseBox to include Feature Extraction and Change Detection [19].

The FuseBox: Land ATD system is being designed to improve the detection of land targets in difficult scenarios (e.g., small targets in high clutter, and those that are camouflaged and/or concealed), by detecting targets in SAR imagery along with either Multispectral Imagery (MSI) or Hyperspectral Imagery (HSI) data. The two detection sets are then fused, which increases the Probability of Detection (P_{det}) and lowers the False Alarm Rate (FAR). It also has a facility for interactive verification of the detection list.

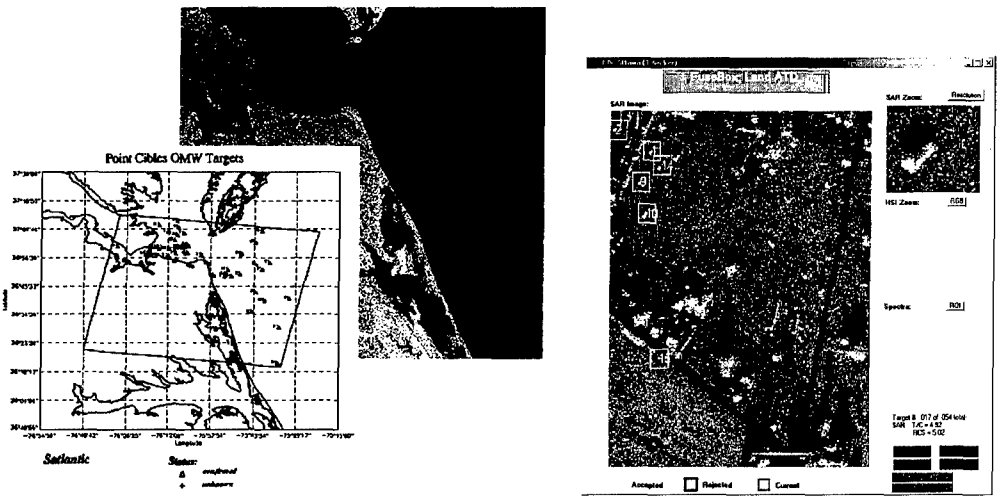

Figure 5: *ATD applications called by the ATR Workbench. (Left) OMW detection results (front) for RADARSAT image (back). (Right) FuseBox: Land ATD detection results.*

There are several algorithms that will be implemented for the target detection and fusion processes. For SAR imagery, two data-based statistical segmentors are currently implemented for target detection. One of these has a CFAR detector, while the other has detection based upon a mean threshold. There are two additional algorithms under development, including a model-based statistical segmentor with a CFAR detector. In all cases, the option to speckle filter the raw SAR image prior to detection will be available. For HSI data, the Spectral Angle Mapping (SAM) algorithm is currently implemented for target detection. For MSI, an anomaly detection algorithm will also be implemented for target detection. For the fusion process, a binary fusion algorithm is currently implemented to fuse the two detection lists. It is expected that another fusion algorithm based upon evidential reasoning will be implemented.

2.4 Preliminaries for the Classification of Ground Vehicles

Since man-made objects are often built with a high degree of symmetry and periodicity that is not prevalent in nature, and since these characteristics are replicated in imagery of such objects, Fourier coefficients have been found to be a useful space from which to derive characteristic feature for target classes of man-made objects [37].

Any image can be decomposed into a combination of suitably weighted periodic functions. When those functions are the complete set of discrete two-dimensional sinusoids supported within the image size, the weights form a set of numbers, the so-called Fourier coefficients, that uniquely describe the image in reciprocal space.

The Fourier coefficients of a $P \times Q$ sized image, I_n^t, can be calculated using the two-

dimensional Discrete Fourier Transform, namely,

$$F_n^t(k_p,k_q) = \sum_{p=1}^{P}\sum_{q=1}^{Q} I_n^t(p,q)e^{i2\pi(pk_p/P^2+qk_q/Q^2)}, \tag{2.4}$$

for each $k_p \in [1,P]$ and $k_q \in [1,Q]$ in reciprocal space. This yields a set of PQ complex coefficients, from which we can extract two numbers: the real and complex values. For real-valued image pixels, e.g. amplitudes, the real part of the coefficients correspond to a cosine decomposition and the imaginary part to a sine decomposition of the image. Since sine and cosine functions are not mutually independent, differing by a constant phase shift of $\pi/2$, we have each real-valued number appearing as a pair. Thus, PQ Fourier coefficients are sufficient to reconstruct the original $P \times Q$ image.

The observed separation and direction of periodicity in an image, however, is a function of the imaging geometry with the object. Both the independent and variation methods indicated in Section 1.2.1 have been examined to best achieve (1.2), and it was determined that eliminating azimuthal and depression angle dependence produced better results, which are implemented as preprocessing algorithms.

To eliminate dependency on the depression angle, θ_d, the slant-range, s_r, is converted to ground range (along a flat-Earth plane), g_r, by projecting onto the ground along a direction perpendicular to the slant plane, i.e., $g_r = s_r/\cos(\theta_d)$ and resizing the pixels in range accordingly. Note that θ_d is independent of the target object and depends only on the geometry of the sensor during data collection. This method recovers the correct ground range distance between two point source targets on flat, level ground when imaged by a radar sensor at infinite distance (in order to have plane waves).

On the other hand, the azimuthal angle does depend on the target object, in the sense that an object lacking rotational symmetry will have a "front" which defines the object heading, ϕ_t, while the sensor motion in cross-range defines the flight heading, ϕ_f. The relative difference provides the azimuthal angle,

$$\phi_a = \phi_t - \phi_f + \phi_s, \tag{2.5}$$

where ϕ_s is the squint angle measured from the aircraft heading to the look direction of the sensor in the clockwise sense. If an object does have rotational symmetry (in the ground plane), then the image need only be rotated to an orientation consistent with any preferred direction that would be part of the class definition, or not at all if the object has circular symmetry. Thus, to mitigate the azimuthal angle effects, it is necessary either to acquire imagery over a complete set of azimuthal angles or to rotate the images to a common orientation. While the former does not require any additional preprocessing of the imagery, it is often not feasible. The latter requires further analysis to determine the object heading in order to apply (2.5).

The current implementation of ATR classifiers is primarily concerned with imagery of ground vehicles. Man-made vehicles typically have a high degree of mirror symmetry about the axis along which the vehicle normally moves, but little symmetry otherwise. Furthermore, stability issues support a preference for vehicles that are longer along the axis

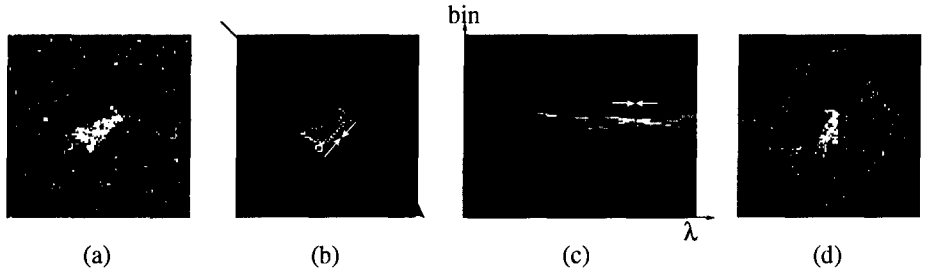

Figure 6: *Orientation of an image of a tank using a Radon transform method. (a) Original SAR image with shadow obscuring right rear of vehicle. (b) Edge binary mask, with far-range shadow line removed. Yellow arrows indicate direction of projection onto the red Radon axis. (c) Radon transform of the edge binary mask. The colors indicate the relative bin values (blue=0, red=max) along the axis corresponding to each* λ*. (d) Image rotated by the* λ *corresponding to the highest bin value.*

of motion than in the other dimensions [38]. Thus, by determining the longest edge of a vehicle in its image representation, ϕ_t can be estimated, up to a rotation of 180°, with a high degree of confidence.

Manual selection of the longest edge can be achieved by allowing the user to draw a line parallel to that edge and the image rotated until the drawn line runs vertically. To automate the process of determining the orientation of an object in a SAR image chip, we implement a Radon transform method [39] on the image. For a given angle, λ, the Radon transform of a digital image projects each pixel onto an axis passing through the centre of the image with slope λ. This property is particularly useful when applied to binary images, where it effectively provides a count of white pixels along each column of the λ-rotated image. Thus, when a line is rotated so as to be vertical, i.e., falling entirely within one column, the Radon transform calculates the length (in pixels) of that line. Since our objective is to find the orientation of the longest edge in an image, the procedure is straight-forward: convert the image into a binary mask of the target pixels, extract the edge pixels and generate the Radon transform of the results for all $\lambda \in [0, 180°]$ (Figure 6). The λ associated with the global maximum value of these transforms will determine ϕ_a up to a rotation of 180°.

This method assumes that all edges of the vehicle will be represented in the image and that the edges of the image representation correspond with actual edges of the object. However, the physics of the radar sensor data acquisition ensures that a shadow region exists, which usually obscures the vehicle's edges in the hemisphere farthest in range from the sensor, as seen in Figure 6a. Often, this shadow region produces an edge to the image representation that runs along a body diagonal of the object, does not correspond to an object edge and is longer than the actual edge of the vehicle. Fortunately, this shadow edge is always oriented in the cross-range direction, and appears along the far-range of the image representation. By removing the far-range pixels of the edge binary mask, this shadow edge can be removed with certainty when present (Figure 6b). If the shadow edge is not present, then this process can destroy the farther edge or edges of the image representation, but the mirror symmetry

assumed for the vehicle assures us that at least one of the longest vehicle edges will survive and will be identifiable using the Radon transform.

Since obtaining accurate orientations for each object is vital for the operation of classifiers designed for azimuthal independence [40, 25], several algorithm variations were experimented with to find the most accurate method. All the variations considered involve extracting the segmentation of the target region from the image chip.

The first method attempted followed these steps: threshold the image, select the connected region with the highest median intensity, take the perimeter pixels then remove the pixels in the shadow region and find the Radon transform as described above. This method gave satisfactory results, although it was found that it was susceptible to errors for images where parts of the object were disconnected due to shadow. As well, the algorithm assumes the maximum intensity region in the image is part of the target.

A more robust method that would include these disconnected target regions was attempted. The method utilized a morphological operation [41] in an attempt to bridge the disconnected regions. First, region-growing was applied to each mask, possibly connecting nearby regions to each other. Then, contraction was attempted, but implemented only where the newly made connections were unaffected. For regions that are located in close proximity and very likely part of the same larger object, this had the effect of connecting regions, while otherwise leaving the boundary edges unchanged. In particular, regions that are partially located in the shadow could often be joined to the primary object.

A third variation was considered that was similar to the second method, but also attempted to make the selection of the target object more robust. Instead of selecting the one region with the highest intensity pixel as the target, all the regions above a specified intensity per number of pixels value were selected. Therefore, all disconnected regions of the target would be included, but this time extra pixels were not created to bridge the regions.

Evaluations of these methods are described in Section 3.2

2.5 The Classifiers

At this point, we have a proposed feature space of Fourier coefficients acting on imagery that have been processed to eliminate, as much as possible, dependency on acquisition geometry. In order to automatically classify an image, we now need to have one or more trained classifiers available. For ground vehicle targets, we initially implement a set of HNeT-based classifiers and an MLP NN classifier. Maritime classifiers are under investigation [24, 42], but are not yet in a suitable state for consideration.

2.5.1 Training and Design Criteria

Currently, training of the classifiers is conducted outside the ATR Workbench, and so the classes available for ATR during operation of the ATR Workbench are limited to the set of

Table 1: *Training images for the standard MSTAR evaluation method. One vehicle (unique serial number) for each class is used, with all imagery having a 17° depression angle, but complete azimuthal coverage, i.e. 0°–360°.*

Targets Type (serial number)	Number of Samples
T-72 (132)	232
BTR-70 (c72)	233
BMP-2 (9563)	233
Total	698

class definitions used for training. Manual classification, however, does permit the user to alter class definitions at will, for example to subdivide a class into different types.

For demonstration purposes, the classifiers have been trained to recognize vehicles belonging to the public release MSTAR data [22]. Here, three target classes have been selected, C_1 is the T-72 main battle tank, C_2 the BTR-70 armored personnel carrier and C_3 the BMP-2 infantry fighting vehicle. Imagery of a single vehicle from each type, collected at $\theta_d = 17°$ and varying ϕ_a, have been selected to form the training set (Table 1). Hence we have, $t_1 \in C_1$, $t_2 \in C_2$ and $t_3 \in C_3$ and a set $\bigcup_{i\in\{1,2,3\}}\{I^{t_i}_{n_i}\}^{N_i}_{n_i=1}$.

For both classifiers, this training data, along with the slicy (a non-vehicle object) imagery from the MSTAR set [22], is pre-processed as indicated above with ϕ_a calculated using the ϕ_t provided for each vehicle based on ground-truth measurements. The total set of images, which class each image belongs to and the Fourier coefficient criteria are used to initialize each classifier training process. Other parameters are set using empirical results from previous analyses [23, 20, 21]. The HNeT training process is optimized for one class definition only, and so three binary classifiers are generated, each trained using the image set of a single target vehicle as the in-class set and all remaining images as the out-class set. The MLP NN, on the other hand, processes a classification node vector for all three classes simultaneously. Once trained, the classifiers can be applied to any image of the correct number and format of pixels (e.g., 64×64 magnitude data). Each classifier produces a real-valued output measuring the similarity to the classifier class. Ideally, that output will be 1 when the target in the image is in-class and 0 when it is not, namely $H_i(t) = \chi_{C_i}(t)$ for each $i \in \{1,2,3\}$. In reality, all we can guarantee is

$$\hat{H}_i\left(f_1(I^t_n), f_2(I^t_n), \ldots, f_M(I^t_n)\right) \approx \hat{\chi}_{C_i}\left(f_1(I^t_n), f_2(I^t_n), \ldots, f_M(I^t_n)\right), \tag{2.6}$$

whenever $I^t_n \in \bigcup_{i\in\{1,2,3\}}\{I^{t_i}_{n_i}\}^{N_i}_{n_i=1}$. For a general image I^t_n, we observe

$$\hat{H}_i\left(f_1(I^t_n), f_2(I^t_n), \ldots, f_M(I^t_n)\right) = \begin{cases} 1+\delta^t_n, & \{f_m(I^t_n)\}^M_{m=1} \sim \{f_m(I^{t_i}_{n_i})\} \\ & \quad \text{for some } n_i \in \{1,\ldots,N_i\}, \\ 0-\delta^t_n, & \{f_m(I^t_n)\}^M_{m=1} \sim \{f_m(I^{t_i}_{n_j})\} \\ & \quad \text{for some } n_j \in \{1,\ldots,N_j\} \\ & \quad \text{with } j \neq i, \\ 0.5 \pm \delta^t_n, & \text{otherwise.} \end{cases} \tag{2.7}$$

for some error term δ_n^t dependent on the degree of similarity (indicated by the $\sim$ function) between I_n^t and the training images. As HNeT uses undisclosed proprietary algorithms, the specific form of the similarity function remains unavailable at this time. Likewise, the understanding of how the highly non-linear node interactions and indeterminate training of Neural Networks is still an active area of basic research in the scientific community.

Since the $\hat{H}_i$ are not characteristic functions and the results on a new image may well be uncorrelated between different $\hat{H}_i$, the question of how to deal with contradictory results leads to methods of classifier fusion and decision theory. Many of these methods assume classifier output in a probabilistic or percentile form, so it is advantageous to map $\hat{H}_i$ to a likelihood factor whose range is the unit interval $(0,1)$. This is achieved by using a sigmoidal function, whereby

$$L_i(I_n^t) = \frac{1}{1+\exp\left(S\left(1-2\hat{H}_i\left(\ldots,f_m(I_n^t),\ldots\right)\right)\right)}, \tag{2.8}$$

where S is a scale factor that determines the extent of linearity for the sigmoid function. Typical values of $S = \ln(19)$ or $S = \ln(99)$ are commonly used.

A further issue arises as to providing a measure for the likelihood that the imaged object belongs to none of the defined classes. Ideally, we have

$$\chi_{C_i}(t_0) = 0, \tag{2.9}$$

for all i whenever $t_0 \notin \bigcup_i C_i$. However, (2.7) indicates that we should expect

$$\hat{H}_i\left(\ldots,f_m(I_n^{t_0}),\ldots\right) \approx \frac{1}{2}, \tag{2.10}$$

for all i and n. This discrepancy is entirely due to the consequences of using a finite set of training data, namely that a finite sequence is *nowhere dense*.

Despite the implications of (2.10), we choose to model (2.9) at present while noting the need for a thorough decision theory analysis of these processes. Thus, for each image, we select the two highest likelihood factors, i.e., we define

$$\begin{array}{ll} i_1, & \text{such that } L_{i_1}(I_n^t) \geq L_i(I_n^t) \text{ for all } i, \\ i_2, & \text{such that } L_{i_2}(I_n^t) \geq L_i(I_n^t) \text{ for all } i \neq i_1, \end{array} \tag{2.11}$$

which we use to generate a likelihood factor for the Unknown pseudo-class according to

$$L_0(I_n^t) = \max\left\{0, 1 - L_{i_1}(I_n^t) - L_{i_2}(I_n^t)\right\}. \tag{2.12}$$

Finally, these likelihood factors are converted into percentiles, so that a normalized result is available,

$$P_i(I_n^t) = \frac{L_i(I_n^t)}{\sum_{j=0} L_j(I_n^t)}, \tag{2.13}$$

which can then be integrated by the ATR Workbench into the information about the target image.

2.5.2 HNeT

AND Corporation's Holographic Neural Technology provides a novel approach to pattern recognition [23, 20]. Large feature spaces are investigated for invariants within the class definitions of the training set data. Feature values are computed for each training sample and transformed into the angular component of complex vectors and these vectors coherently combined. Features that are relatively constant within a training class reinforce, while randomly valued features average to zero. The most invariant of these features are stored and used as the baseline against which the similarity of test data is measured.

Thus, HNeT is able to rapidly investigate large numbers of potential features in order to determine the most invariant. Since a wider space of potential features is considered compared to most other methods, which usually require specific features to be predetermined, classifiers based on HNeT will usually be using feature sets that are more characteristic of the class definitions involved, and are therefore likely to produce better results.

In this case, features based on the entire space of 4096 Fourier coefficients for 64px $\times$ 64px magnitude images are investigated. Three classifiers, one for each of the T-72, BTR-70 and BMP-2 classes, are applied, each optimized to determine the 256 most invariant features among the training images that belong to the associated class. Each HNeT classifier will generate a measure of similarity to the values of these baseline features. Images exhibiting similar values to the in-class training data for the selected features of each classifier will have an HNeT result near 1. Those that do not will be closer to 0. Figure 7 graphically shows the HNeT results for a T-72 classifier applied to the MSTAR test data.

These trained HNeT classifiers have been incorporated into the ATR Workbench as a classifier module that will generate three similarity values for each image. The ATR Workbench converts these into likelihood factors via (2.8) and uses (2.12) to generate a measure for belonging to none of the classes.

2.5.3 MLP Neural Network

A Multi-Layer Perceptron Neural Network is able to make decisions based on learning patterns from sample data. The MLP NN is a supervised network, meaning it requires knowledge about the correct target output in order to learn. The NN is trained outside the ATR Workbench using a standard back-propagation training method, the delta rule or Widrow-Hoff algorithm [43]. The training process is an iterative manipulation of the weights of the network connections, and usually continues until all the training samples, I_n^t, are correctly classified to their own type. This means the values of the NN node vector for each sample will all be within a globally specified error term, ε, of the desired vector values, i.e.

$$NN(I_n^t)_i - (D_n^t)_i < \varepsilon, \tag{2.14}$$

for each vector element, i, with

$$(D_n^t)_i = \begin{cases} 1, & I_n^t \in C_i, \\ -1, & \text{otherwise.} \end{cases} \tag{2.15}$$

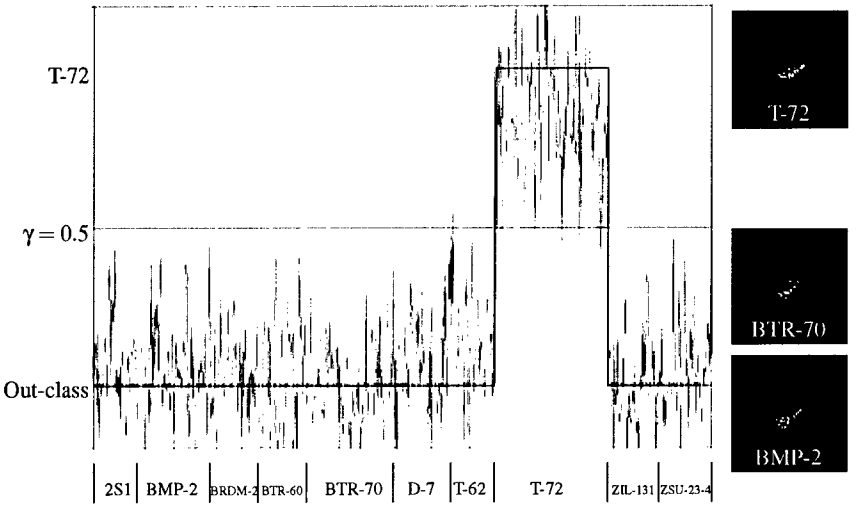

Figure 7: *HNeT results for a T-72 classifier applied to the MSTAR 15° depression angle test data. For each image located along the horizontal axis, the corresponding HNeT output is given by the red peak. The blue line indicates desired results. With the threshold set to* $\gamma = 0.5$*, a peak above* γ *is classified as T-72, while a peak below is not.*

However, because Neural Networks are highly non-linear, the training process is an indeterminate method, which therefore does not guarantee that the algorithm will avoid local minima and achieve (2.14) for any given training attempt. If convergence does not occur within a predetermined time, the training process is reinitiated. The state of the NN is frozen after the learning process is completed, the network weights are no longer altered. Should a new class be added to this system, the NN will have to be completely re-trained with that new class included.

For demonstration purposes, a MLP NN application was developed for the MSTAR public dataset. It was configured as illustrated in Figure 8 with 13 nodes in the first hidden layer, 11 nodes in the second hidden layer, 38 nodes in the input layer and 3 nodes in the output layer. The NN is used to classify the three military ground vehicles, a BMP-2, BTR-70 and T-72, requiring an output layer with three nodes. Node 1 is assigned to T-72, Node 2 assigned to BTR-70 and Node 3 assigned to BMP-2. As with the HNeT classifier, the standard MSTAR training data (Table 1) was used to train the MLP classifier. Successful training time was observed to vary over a large period, the majority ranging from 5 to 30 minutes.

The MLP NN requires considerably more computation resources than HNeT for this type of application. To reduce the size of the input dimension, only 16 coefficients per class are selected, those that best separate the decision space between the three classes. Of these 48 coefficients, only 38 are unique for the MSTAR training set, which is the reason 38 nodes are selected for the input layer. For training, the desired vectors are $(1,-1,-1)$, $(-1,1,-1)$

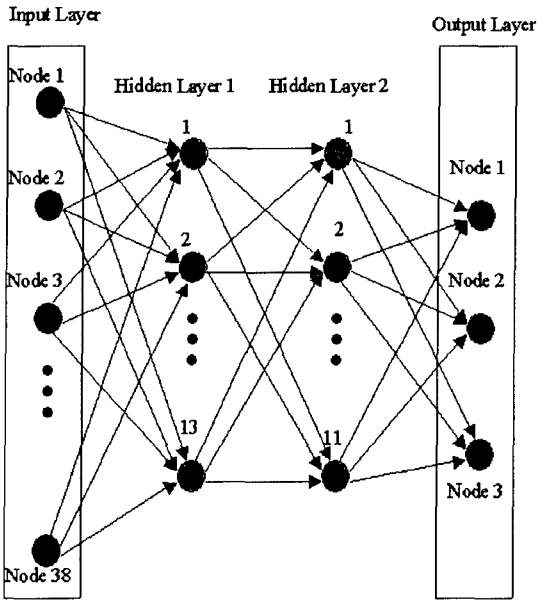

Figure 8: *Multi Layer Perceptron Neural Network*

and $(-1,-1,1)$ for the T-72, BTR-70 and BMP-2 vehicles, respectively. For operational use, the output results are subjected by a unit shift and halved, so that the ideal values will be mapped to either 0 or 1.

This trained MLP NN has been integrated into the ATR Workbench as a classifier module. A likelihood measure that the object belongs to none of the defined classes is calculated using (2.12) and output values are converted into likelihood factors using (2.8).

3 Results

3.1 Detection

The FuseBox: Land ATD system was used outside the ATR Workbench to analyze a MSTAR clutter scene with target chips inserted (i.e., Demoscene.bin, 1476 cols by 1784 rows) [19]. The analysis of this single-epoch SAR image was simplified from Section 2.3.2 description for a multi-sensor case, and it consisted of the following steps:

a. Detection of targets on the raw SAR imagery (no despeckling), using the data-based CFAR segmentor. The parameters were:

   (a) Window size = 5 pixels
   (b) Threshold = 0.85
   (c) CFAR = 10^{-6}
   (d) Segment size thresholding, with minimum segment size equal to 20 pixels, and

Table 2: Orientation Algorithm Variations Results

Algorithm Variation	Average Error	Standard Deviation
First - Independent	3.9°	6.4°
Second - Morphological Bridging	2.0°	5.9°
Third - Region Pixel Intensity	3.8°	6.5°

maximum equal to 500 pixels.

b. Reduction of false alarms using the interactive verification window as described in Section 2.3.2, to accept or reject the detected targets.

Of the 54 targets initially detected on the MSTAR image, 14 remained after the interactive verification stage. Targets that were deleted during this stage were primarily multiple detections of obviously extended (clutter) objects and image defects. Note that the segmentation process used in this analysis renders a border region (i.e., 2 pixels, related to the window size) around the edge of the image unusable for detection.

3.2 Pre-Processing Algorithm Variations

The image chip pre-processing algorithm variations (described in Section 2.4) were each evaluated with approximately 700 images from the MSTAR public dataset. Once again the BMP-2, T-72 and BTR-70 vehicle types were used, each vehicle being imaged from varying azimuthal directions. The algorithm results were compared with the known image angle, and the performance statistics calculated, as presented in Table 2. These evaluations were conducted external to the ATR Workbench in order to determine which would be implemented. For the first algorithm, the angle differed from the correct value by 3.9° on average with a standard deviation of 6.4°. The second algorithm, which used the morphological bridging operation, had an average error of 2.0° and a standard deviation of 5.9°. The final algorithm variation, using the intensity per number of region pixels had an average error of 3.8° and a standard deviation of 6.5°.

From this analysis the morphological bridging method was the most accurate and has, therefore, been implemented in the ATR Workbench for calculating target vehicle orientation. The original algorithm suffered from not including disconnected parts of the target, particularly regions in the shadow, while the third variation suffered from extra non-target regions being included that would cause more errors.

3.3 Classifier Performance

The HNeT and MLP NN classifiers have been trained on the public release MSTAR data and integrated with the ATR Workbench. Both these classifiers have shown good performance, as determined by standard methods for evaluating the performance of a classifier, which include the use of Confusion Matrices and the generation of Receiver Operation

Table 3: *Confusion Matrices ($P_{det} = 0.9$) for the MSTAR baseline, MLP Neural Net and HNeT classifiers. Imagery at 17° depression angle of the vehicles indicated by asterisks are used for training (see Table 1). All test imagery have 15° depression angles.*

MSTAR	BMP-2	BTR-70	T-72	Other
BMP-2 (9563)	**153**	18	4	*21*
BMP-2 (9566)	**143**	36	6	*11*
BMP-2 (c21)*	**187**	5	0	*4*
BTR-70 (c71)*	4	**188**	0	*4*
T-72 (132)*	5	5	**173**	*13*
T-72 (812)	19	1	**111**	*64*
T-72 (s7)	19	10	**143**	*19*
			$P_{cc\|d}$= **0.8934**	

MLP NN	BMP-2	BTR-70	T-72	Other
BMP-2 (9563)	**137**	2	28	*28*
BMP-2 (9566)	**148**	5	20	*23*
BMP-2 (c21)*	**157**	2	15	*22*
BTR-70 (c71)*	2	**187**	3	*4*
T-72 (132)*	4	1	**187**	*4*
T-72 (812)	42	0	**119**	*34*
T-72 (s7)	48	11	**117**	*15*
			$P_{cc\|d}$= **0.8518**	

HNeT	BMP-2	BTR-70	T-72	Other
BMP-2 (9563)	**181**	0	4	*10*
BMP-2 (9566)	**164**	2	12	*18*
BMP-2 (c21)*	**191**	1	0	*4*
BTR-70 (c71)*	0	**194**	0	*2*
T-72 (132)*	0	0	**193**	*3*
T-72 (812)	26	2	**115**	*52*
T-72 (s7)	8	8	**134**	*41*
			$P_{cc\|d}$= **0.9490**	

Characteristic (ROC) curves. The standard evaluation suggested by DARPA/WL applied to the MSTAR data set provides a baseline with which to compare. The HNeT and MLP NN classifiers were applied to the MSTAR data set and then compared with the MSTAR baseline classifier. Using the multi-vehicle data for the BMP-2, BTR-70 and T-72 classes present in the MSTAR data set, the classifiers were trained on imagery at 17° depression angle for a single vehicle in each class and then tested on imagery at 15° depression angle for all the vehicles in each classes. These evaluations were done outside the ATR Workbench.

The detailed HNeT evaluation, including a comparison to the MSTAR baseline is described in [20]. A detailed description of the MLP NN performance is presented in [21], but it's attributes have not previously been compared with any other classifier. Information about the trained vehicle types and their serial numbers, as well as evaluated vehicle types and their serial numbers, are given in [20] and [21]. For quick review, we have included the confusion matrices (Table 3) and ROC curves (Figure 9) for the MSTAR baseline, HNeT and MLP NN classifiers.

The confusion matrices show the number of correctly classified images and overall percentage of correct classification ($P_{cc|d}$) for each of the classifiers. The operating point of the confusion matrix for the MSTAR baseline, MLP NN and HNeT is selected at P_d = 0.9.

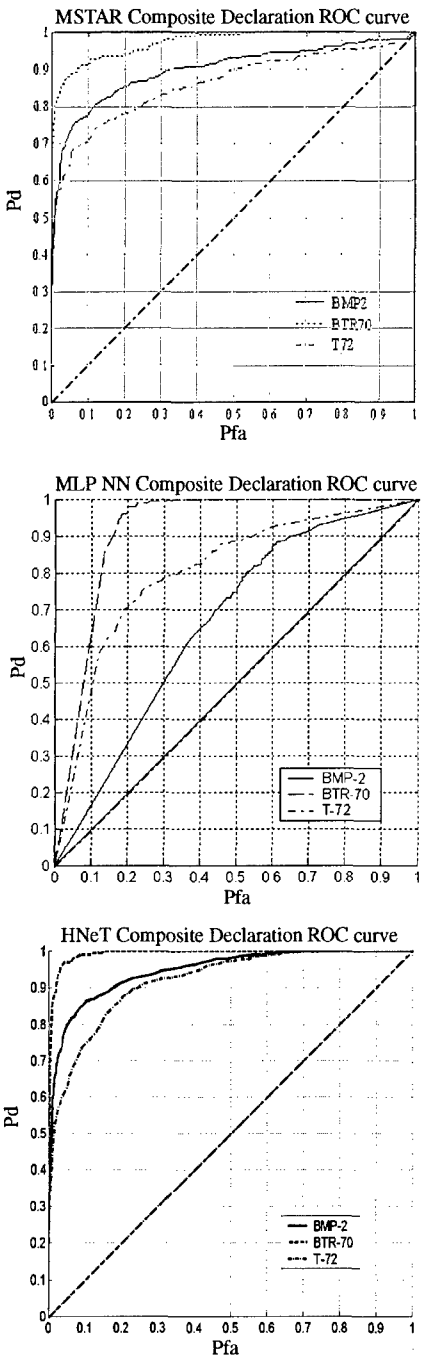

Figure 9: *Receiver Operating Characteristic curves for the MSTAR Baseline, MLP Neural Net and HNeT classifiers.*

Correct classification is indicated in bold. Overall $P_{cc|d}$ is 89% for the MSTAR baseline, 95% for the HNeT and 85% for the MLP NN. The MSTAR method has both BMP-2 and T-72 images being misclassified as BTR-70s, while the HNeT and MLP NN results show them being confused with each other, and rarely with the BTR-70. The major cause for this is expected to be the difference in features used, since the MSTAR baseline is a template matching algorithm, while HNeT and the MLP NN use Fourier coefficients. Furthermore, HNeT uses 256 coefficients compared to 16 features for the MLP NN, which accounts for much of the difference between the two.

When producing the ROC curves, the MSTAR baseline, HNeT and MLP NN used the same confusers (2S1 self-propelled howitzer and D7 bulldozer). The ROC curves in Figure 9 show the relationship between percentage of declared targets and percentage of false alarm. The false alarm increases with increase of detecting targets. The diagonal lines in Figure 9 are the results for a random classifier, where the output class is completely independent of the type of image given as input. Better classifiers should provide lower P_{fa} rates and higher P_d rates, i.e., an ideal operating point is the left upper corner $(P_d, P_{fa}) = (1,0)$. We can see that all the classifiers are better than the random classifier. However, the HNeT results clearly show a better performance than either the MSTAR baseline or MLP NN classifier.

4 Demonstration

To demonstrate the main features of the ATR Workbench, the following script has been designed for use in real-time presentations of the software.

4.1 Outline

After introducing the topic of ATR, the motivation and the goals for the project the ATR Workbench software is started. An image that has been specifically created for the demonstration is loaded (Figure 10). The scene consists of a real SAR image with 10 MSTAR target chips manually inserted at plausible locations. Prior to insertion, the pixel values of

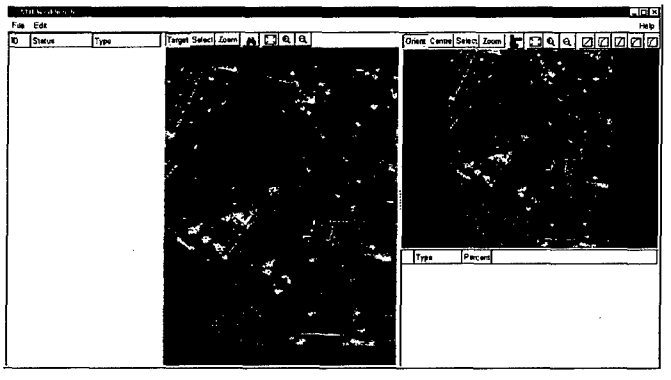

Figure 10: *Demo: new image loaded into the ATR Workbench.*

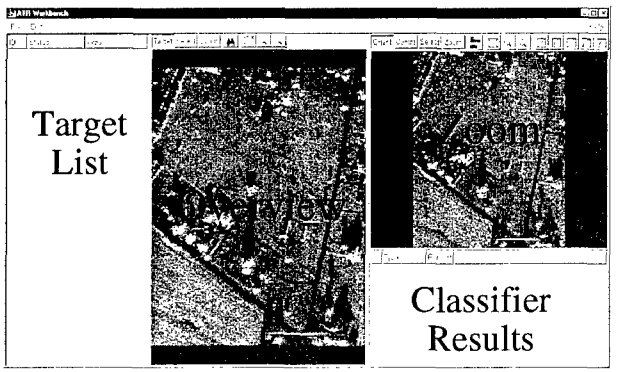

Figure 11: *Panel function overview.*

the chips were altered to equalize the clutter statistics of the chip to that of the scene pixels being replaced. The chips were placed as two groups, each consisting of a T-72 MBT, a BMP-2 IFV, a BTR-70 APC, a 2S1 SPH and a D-7 bulldozer.

There are four main panels (Figure 11) in the ATR Workbench application as described in Section 2.2.3. The target list provides a text listing of all the identified targets, with their current status. The scene overview panel gives a high-level view of the image with targets marked by coloured boxes representing their type (if known). The zoom panel shows a detailed view of the currently selected target and provides a method to manually orient it if necessary. Finally, the classifier result panel displays likelihood measures generated by the classifier that a target is of a particular type.

4.2 Operation of the ATR Workbench

Panning and zooming in the scene overview window can be accomplished with the scroll bars at the edge of the panel and the zoom buttons at the top. After examining the scene a potential target is identified. Switching to target mode occurs by using the "Target" button at the top of the panel and then drawing a box around the target in the image (Figure 12). While in target mode, the box can be resized. Changing to "Select" mode, the target box can

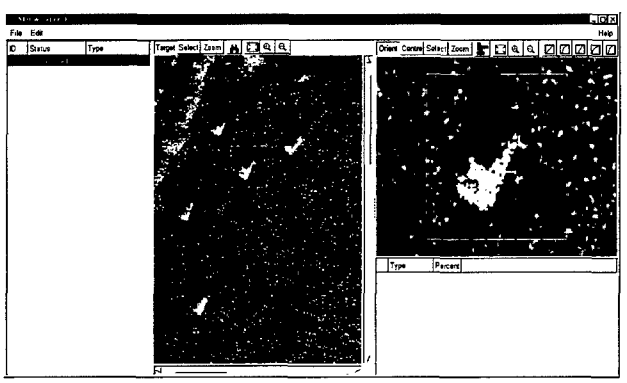

Figure 12: *Demo: target selection.*

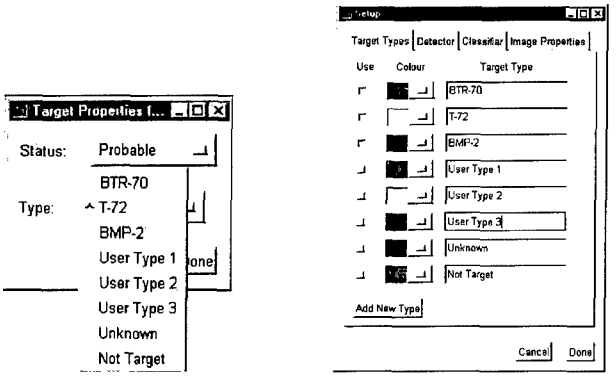

Figure 13: *Demo: (left) target type selection menu, (right) class type setup menu.*

be moved elsewhere in the image. Finally, pressing the "Reset" button centres the image and adjusts the zoom so that the entire scene is visible in the view.

The newly created target will be added to the target list on the far left. By selecting it from the list, the target becomes highlighted in the overview panel and is displayed in the zoom window. The zoom window has several different visual enhancement modes that can be selected using the buttons at the top of the zoom window. The target can be manually classified with a right-click on the item in the target list, and then selecting the appropriate target type from the drop-down list (Figure 13).

The File/Setup menu lets the user define the list of target types that appear in the manual classification drop-down list. As well, colours can be associated with individual target types and new target types can be defined. The Setup dialog also has tabs for the detector, classifier, and image properties. Each of these has custom settings that can be adjusted.

Pressing the "binocular" button at the top of the overview panel imports a target list created by an external ATD application (Figure 14). In this case, a simulated application of the FuseBox: Land ATD detector has previously generated a target file for the demonstration image that is ready for use by the ATR Workbench. Once ingested, each potential target is marked in the overview panel with a green box and a corresponding entry is displayed in

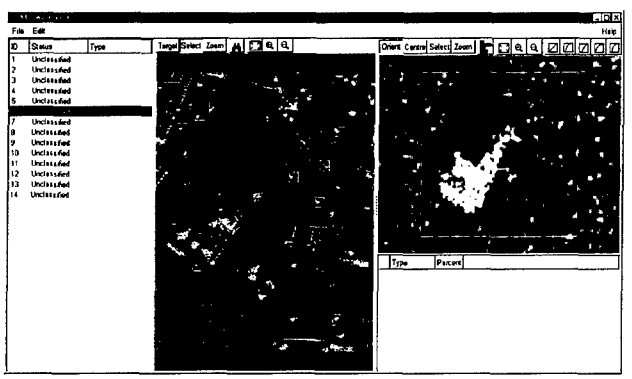

Figure 14: *Demo: importing a target list from an external ATD application.*

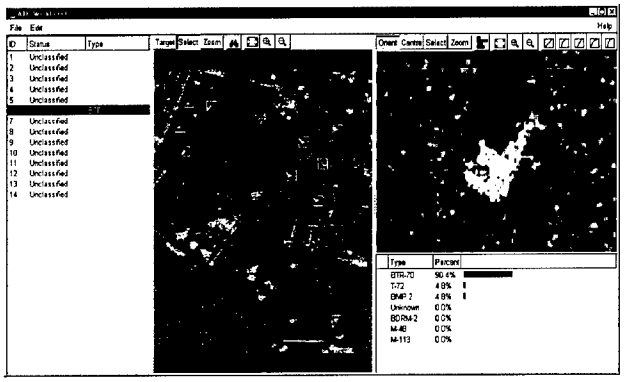

Figure 15: *Demo: displaying ATR module classification results.*

the target list.

When a target is selected, either from the target list on the left or by using "Select" mode and clicking on a target box in the overview panel, classification by the currently active ATR module can be initiated via the "bar graph" button at the top of the zoom window. Either the HNeT or the MLP NN classifier will be used depending on the current selection in the Setup parameters. After the classifier has run, the likelihood estimations will be presented in the bottom right result panel (Figure 15). As well, the target status will be assigned "classified" in the target list with the most likely type indicated. In the overview panel the colour of the marker box will be updated to reflect the new status.

The classified results can be manually overridden by selecting the target in the target list and right-clicking to bring up the status dialog. By changing the status to "probable" or "possible" the type can be manually selected. If the target type is returned to "classified" then the type will be automatically assigned according to any classifier results associated with the target.

In the zoom window, selecting the "orient" mode allows the user to draw a line in the zoom window to indicate the orientation of the target (Figure 16). The window will then rotate to the new orientation so that line, and presumably the target, appears vertical in the window.

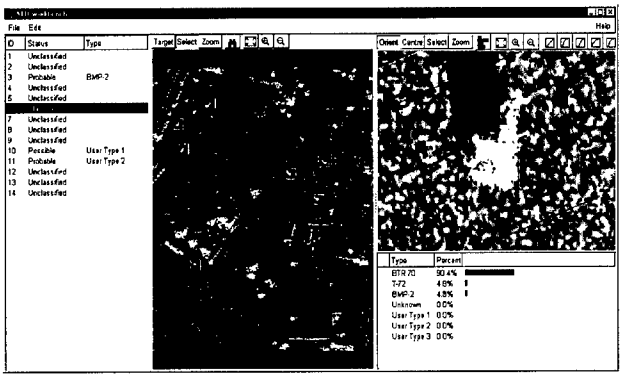

Figure 16: *Demo: orienting an image chip.*

Table 4: *Sample report file showing manual and automated ATR Workbench analysis on target list generated by simulated application of FuseBox: Land ATD. The Percentage List has been truncated here for display purposes.*

ID	Status	Type	Classifier	Percent List
1	Unclassified			
2	Unclassified			
3	Probable	BMP-2		
4	Unclassified			
5	Unclassified			
7	Classified	BTR-70	MLP-Neural-Net	(BTR-70=90.4%, T-72=4.8%, ...)
8	Unclassified			
9	Possible	User Type 1		
10	Probable	User Type 2		
11	Unclassified			
12	Unclassified			
13	Unclassified			

The second classifier can be used by selecting the menu item File/Setup/Classifier and choosing it from the drop down list. The new classifier will be initiated whenever the classification button at the top of the zoom window is pressed.

Report files can be generated using the menu File/Report (Table 4). An online help system is also available from the Help menu, launching an HTML web browser pointed at user documentation for the system.

5 Conclusions

5.1 Summary

A software tool linking target classifiers with target detectors for the analysis of SAR imagery has been developed, enabling interactive or automated application of each step of the target recognition process. This so-called ATR Workbench works upon target lists, either generated interactively or imported from ATD software, and modularly integrates currently available ATR software with intermediate processing algorithms through a user interface that will allow target selection, orientation, segmentation and classification to be implemented by the user or by one or more automation modules.

The ATR Workbench is designed using OpenEV libraries and the Python programming language to facilitate integration of new modules and interfacing with external applications, such as the OMW ship detector, the HNeT classification engine and various Matlab-based components. The ATR Workbench itself can easily be ported to most popular platforms and operating systems.

As an R&D tool, this software is in a state that will allow, amongst other things,

a. current target recognition methodologies used by the Canadian Forces to be examined in terms of the impact of automation,
b. the development of standardized empirical evaluation techniques for SAR ATR applications in terms of effectiveness on Canadian Forces capabilities,
c. evaluation of specific automation algorithms for SAR ATR, both current and future, in terms of effectiveness,
d. better exposure of operational personnel to various automation capabilities of the target recognition process as part of demonstrations of end-to-end ATR systems,
e. a baseline for the operational community to provide feedback on further research, development and implementation issues concerning SAR imagery analysis and exploitation,
f. opportunities for operational personnel to develop confidence in the application of specific SAR ATR algorithms to specific operational tasks.

Further development of the software is expected to allow the ATR Workbench to be configured to provide the prototype for any number of future ATR systems that could be implemented by the Canadian Forces to enable automation for operational target recognition requirements.

5.2 Future Work

The future direction of the ATR Workbench will depend largely on the feedback received from the Canadian Forces on how the system can better suit their requirements and what capabilities are of greatest importance to them. There are, however, a number of features that could certainly be of benefit that were not included in the initial version of the ATR Workbench due to project constraints.

Currently, the training required for HNeT and the MLP Neural Net classifiers is performed outside of the ATR Workbench using a process that is manually intensive. The implementation of a training mode in the ATR Workbench would help to speed up the training process and thereby facilitate experimentation with more numerous target classes.

The ATR Workbench GUI only has one zoom window to display the currently selected target, not the three windows envisioned in the original concept. Additional windows would provide the ability to compare and analyze several targets simultaneously.

Further work will almost certainly involve integrating additional target classifiers and interfacing detectors to the ATR Workbench for experimentation and evaluation, as this continues to be one of the primary objectives of the ATR Workbench project.

Prior to a target being classified, the pre-processing module automatically segments the target from the image. Although the segmentation algorithm works reasonably well in most circumstances, the user may want to override the algorithm and manually adjust the target segment. This would bring the target segmentation tool in line with all other tools in the ATR Workbench, in that it would have both an automated and manual method.

Finally, the imagery used in the ATR Workbench is single channel SAR. Other types of single channel imagery (IR, optical) could be processed by the ATR Workbench if supported by appropriate detectors and classifiers. As well, future work to support multiple channel data, such as polarimetry or a fusion of multisensor imagery, has the potential to improve the classifier results.

References

1. A. Garvey and V. Lesser (1994). A survey of research in deliberative real-time artificial intelligence. *Real-Time Systems*, **6**, 317.

2. H. Rothe (1998). Approaches to Pattern Recognition. In *Advanced Pattern Recognition Techniques*, NATO/RTO EN-2. p. 1-1.

3. S. Mori, C. Y. Suen and K. Yamamoto (1992). Historical Review of OCR Research and Development. In *Proc. of the IEEE, Vol. 80*, p. 1029.

4. K.L. Sala (1997). Image Classification by Neural Networks Using Moment Invariant Feature Vectors. (CRC Report No. 97-002). Ottawa: Communications Research Centre, Industry Canada.

5. S. Young (1996). A review of large-vocabulary, continuous-speech recognition. *IEEE Signal Processing*, **15**, 45.

6. D.J. Burr (1988). Experiments on neural net recognition of spoken and written text. *IEEE Trans. Acoustics, Speech and Signal Processing*, **36**(7), 1162.

7. C. Bunney (1997). Survey: Face recognition systems. *Biometric Technology Today*, **5**, 8.

8. A.S. Tolba and A.N. Abu-Rezq (2000). Combined Classifiers for Invariant Face Recognition. *Pattern Analysis and Applications*, **3**(4), 289.

9. C. Oliver and S. Quegan (1998). Understanding Synthetic Aperture Radar Images, Norwood, USA: Artech.

10. United States Congress: House Permanent Select Committee on Intelligence (1996). IC21: Intelligence Community in the 21st Century: Staff Study, Washington, USA: U.S. G.P.O.

11. T. Pavlidis (1992). Why Progress in Machine Vision Is So Slow. *Pattern Recognition Lett.*, **13**, 221.

12. R. Jain and T. Binford (1991). Ignorance, Myopia, and Naivete in Computer Vision Systems. *CVGIP: Image Understanding*, **53**(1), 112.

13. R. Haralick (1986). Computer Vision Theory: The Lack Thereof. *Computer Vision, Graphics, and Image Processing*, **36**, 372.

14. T.D. Ross, J.J. Bradley, L.J. Hudson and M.P. O'Connor (1999). SAR ATR: So What's the Problem? An MSTAR Perspective. In *Proc. SPIE, Vol. 3721, Algorithms for SAR Imagery VI*, p. 662. Orlando, USA.

15. K.W. Bowyer and P.J. Phillips (1998). Overview of Work in Empirical Evaluation of Computer Vision Algorithms, Los Alamitos CA, USA: IEEE Computer Society.

16. J.M. Power and L.E. Cochran (1999). Operational Digital Imagery Navigator (ODIN) Information System Architecture Investigation. (FMT-WW71460025-01). Ottawa: FirstMark Technologies Ltd.

17. M.D. Henschel, R.B. Olsen, P. Hoyt and P.W. Vachon (1997). The Ocean Monitoring Workstation: Experience Gained with RADARSAT. In *Proceedings of Geomatics in the ERA of RADARSAT (GER'97)*, Ottawa, Canada.

18. Satlantic Inc. (2002). Satlantic Products: OMW. (Online) Satlantic Inc. `http://www.satlantic.com/products/surveillance/omw/` (October 31, 2002).

19. J. Secker (2002). Private communication.

20. R.A. English (2001). Automatic Target Recognition Using HNeT. (DREO TM 2001-080). Defence Research Establishment Ottawa.

21. N.M. Sandirasegaram (2002). Automatic Target Recognition in SAR Imagery using a MLP Neural Network. (DRDC Ottawa TM 2002-120). Defence R&D Canada – Ottawa.

22. T.D. Ross, S.W. Worrell, V. Velten, J.C. Mossing and M.L. Bryant (1998). Standard SAR ATR Evaluation Experiments Using the MSTAR Public Release Data Set. In *Proc. SPIE, Vol. 3370, Algorithms for SAR Imagery V*, p. 566. Orlando, USA.

23. J. Sutherland (1999). SAR Automatic Target Recognition via a Holographic/Quantum Neural Net. (Contract W7714-8-0200/001/SV). Toronto: AND Corporation.

24. A. Damini, C. McMillan, M. McDonald, R.A. English, B. Balaji and G. Haslam (2002). R&D for Maritime Patrol Aircraft Radar. (DRDCO TM 2002-093). Defence R&D Canada – Ottawa.

25. R.O. Duda, P.E. Hart and D.G. Stork (2001). Pattern Classification, 2 ed. New York, USA: Wiley.

26. W. Siedlecki and J. Sklansky (1988). On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, **2**(2), 197.

27. C.M. Bishop (1995). Neural Networks for Pattern Recognition, Oxford, UK: Oxford.

28. Python Software Foundation (2002). Python Language Website. (Online) Python Software Foundation. `http://www.python.org` (October 29, 2002).

29. Atlantis Scientific Inc. (2002). OpenEV. (Online) Open Source Development Network. `http://openev.sourceforge.net/app/files.html` (October 31, 2002).

30. Atlantis Scientific Inc. (2002). OpenEV. (Online) Open Source Development Network. `http://openev.sourceforge.net` (October 29, 2002).

31. G. Walter, F. Warmerdam and P. Farris-Manning (2002). An Open Source tool for Geospatial Image Exploitation. In *IEEE International Geoscience and Remote Sensing Symposium and the 24th Canadian Symposium on Remote Sensing, Vol. 6*, p. 3522.

32. M.M. Taylor, J.G. Hollands, Z. Jacobson, L. Rasmussen, A. Kaster, and M. Varga (2001). Visualisation of Massive Military Datasets: Human Factors, Applications, and Technologies. NATO. Hull, Canada.

33. W. Buxton (1989). On the Road to Brighton. *SIGCHI Bulletin*, **20**(4), 16.

34. ISO 9241 (1987). Ergonomic requirements for office work with visual display terminals.

35. C. Lewis and J. Rieman (1994). Task-Centered User Interafce Design: A Practical Introduction. (Online) HCI Bibliography: Human-Computer Interaction Resources. `http://hcibib.org/tcuid` (October 31, 2002).

36. P.W. Vachon, S.J. Thomas, J. Cranton, H. Edel, M.D. Henschel (2000). Validation of Ship Detection by the RADARSAT Synthetic Aperture Radar and the Ocean Monitoring Workstation. *Canadian Journal of Remote Sensing*, **26**, 200.

37. A.K. Jain (1989). Fundamentals of Digital Image Processing, Englewood Cliffs, USA: Prentice-Hall.

38. H. Ashley and M. Landahl (1985). Aerodynamics of wings and bodies, New York: Dover.

39. P. Toft (1996). The Radon Transform — Theory and Implementation. Ph.D. thesis. Technical University of Denmark. Kongens Lyngby, Denmark.

40. T.M. Caelli and Z.Q. Liu (1988). On the minimum number of templates required for shift, rotation and size invariant pattern recognition. *Pattern Recognition*, **21**(3), 205.

41. R. Haralick and L. Shapiro (1992). Computer and Robot Vision, Volume I, Reading, Massachusetts: Addison-Wesley.

42. Y. Tessier (2000). Evaluation of Decision Rule Based Neural Net on SpotSAR Imagery of Military Ships. (990001532). Montréal: Lockheed Martin Canada.

43. R.D. Reed and R.J. Marks II (1999). Neural Smithing, Cambridge, USA: MIT Press.

44. The GIMP Toolkit. `http://www.gtk.org/` (October 26, 2001).

45. SUN Microsystems, Inc.. Java Advanced Imaging in Action: ESRI, Inc.. (Online) SUN Microsystems, Inc. `http://java.sun.com/products/java-media/jai/inaction/esri.html` (October 26, 2001).

46. A.C. Williams and B. Clark (1996). Evaluation of SAR ATR. In *Proc. SPIE, Vol. 2755, Signal Processing, Sensor Fusion and Target Recognition V*, p. 36. Orlando, USA.

47. J.C. Mossing and T.D. Ross (1998). An Evaluation of SAR ATR Algorithm Performance Sensitivity to MSTAR Extended Operating Conditions. In *Proc. SPIE, Vol. 3370, Algorithms for SAR Imagery V*, p. 554. Orlando, USA.

48. L.M. Novak (2001). SAR Automatic Target Recognition Using Synthetic Aperature Radar, Bellingham WA, USA: SPIE.

49. C. Sadowski (2001). Measuring Combat Identification: A Warfighter Perspective. In *Proc. First Annual ATR Theory Workshop*, Dayton, USA.

Acronyms

AFRL	Air Force Research Laboratory
APC	Armoured Personnel Carrier
ATD	Automatic Target Detection
ATR	Automatic Target Recognition
CF	Canadian Forces
CFAR	Constant False Alarm Rate
COTS	Commercial Off-the-Shelf
DRDC	Defence R&D Canada
EO	Electro-Optical
FAR	False Alarm Rate
GIAs	Geospatial Image Analysts
GIMP	The GNU Image Manipulation Program
GTK	The GIMP Toolkit
GUI	graphic user interface
HNeT	Holographic Neural Technology
HSI	Hyperspectral Imagery
HTML	Hypertext Markup Language
IFV	Infantry Fighting Vehicle
IR	Infrared
LoS	Line of Sight
MBT	Main Battle Tank
MLP	Multi-Layer Perceptron
MSI	Multispectral Imagery
MSTAR	Moving and Stationary Target Acquisition and Recognition
NN	Neural Network
ODIN	Operational Digital Imagery Navigator
OMW	Ocean Monitoring Workstation
R&D	Research and Development
ROC	Receiver Operation Characteristic
SAM	Spectral Angle Mapping
SAR	Synthetic Aperture Radar
SPH	Self-Propelled Howitzer

Annex A
ATR Workbench Functional Design

A.1 Introduction

This document is intended to capture how the ATR Workbench tool will function. It describes the flow of data, the components of the system, the system outputs and how the user can interact with the interface. Based on this functional description we arrive at a set of requirements that the technological environment will need to meet.

This document is intended to facilitate discussion of the ATR Workbench and common tool architecture. Upon the selection of the technology a detailed design for the ATR tool will be completed.

A.2 System Overview

The ATR Workbench system will consist of three components all linked together through a common graphic user interface (GUI). The components are the Ocean Monitoring Workstation (OMW), a custom target orientation system, and HNeT, a Holographic/Quantum Neural classifier system. The GUI will be responsible for calling each of these sub-systems at the appropriate times and passing the data between the subsystems. The GUI will provide the user with the ability to over-ride the results of automated tasks.

A.2.1 OMW

Given an input scene OMW will identify potential targets. Once a target has been identified it will provide the bounding co-ordinates for the target in the scene and extract the image chip containing the target.

A.2.2 Target Orientation/Centering

Given an image chip it determines the orientation of the target. It then rotates the target so that it is vertical. It also attempts to center the target. The output image chip is cropped to dimensions that are a power of two with the target vertical and centered. This step may include additional image processing. Steps can be optional turned on or off by the user.

A.2.3 HNeT

HNeT must have previously been trained with the target types. It takes oriented, centered, power of two, image chips and returns the probabilities for the various target types.

A.2.4 Graphical User Interface (GUI)

The GUI links the above components together into a complete system. It invokes the subsystems in order and provides the appropriate input data and parameters. The results of the

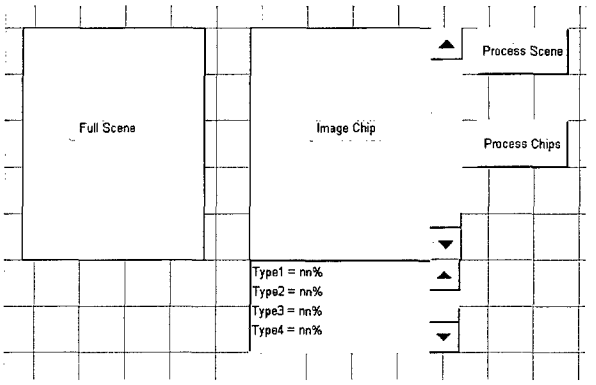

Figure A.1: *GUI Prototype Design*

processing are displayed graphically for the operator to evaluate. As well, it provides the operator with the ability to override the results from each step. This includes the segmentation of the scene into chips, target orientation and target centering.

GUI Features:

- select target in Full Scene window and have it displayed in Image Chip view
- in Image Chip view draw vertical line and centre point on target
- display probability of target type in list below Image Chip and highlight most likely
- file menu to open/close project and load new scene
- Full Scene and Image Chip view have display property controls (max/min range, zoom)
- Rotate Image Chip
- Print report of current results
- Save/Load project state and images

More specifically the tool will perform the following tasks in order:

a. Load Scene (File, load scene) - load scene image into Full Scene view
b. Process Scene Button (Part I)- call OMW with scene as input, return target locations and chips. Highlight targets in Full Scene view.
c. Process Scene Button (Part II) - call Matlab orientation and center function to rotate chips. Display chips in Image Chip view.
d. User Update Target locations - user can highlight and modify bounding region around target in Full Scene view and see updated chip in Image Chip view
e. User Update Target Orientations - user can draw vertical line over target in Image Chip view. Matlab orientation function will automatically be called with new vertical line and perform rotation. Image Chip view will update with new rotation.
f. User Update Target Centre - user can specify point on target in the Image Chip view to indicate the target centre. Image Chip view will update with new centre.
g. Process Chips - call HNeT with image chips as input, return probabilities of target types. Display probabilities for selected chip in list under Image Chip view.

h. User can return to steps 4, 5 and 6 in an attempt to achieve better results.

A.3 System Requirements

Given the above design the selected technology will be required to support:

- OMW and HNeT data
- calling external batch processes (OMW and HNeT)
- raster graphics support (full scene - zoom, pan, histogram)
- vector graphics support (user target center/vertical)
- platform independent
- appropriate for operational environment
- GUI Support

Annex B
Technology Evaluation for ATR Workbench and Toolkit

B.1 Introduction

The functional description of the ATR Workbench gives us a set of requirements that the technological environment will need to meet. Four technologies are proposed and evaluated based on the ATR tool's requirements. As well, the technologies will be evaluated based on general requirements to support a common architecture for a larger image processing toolkit developed by the SAR ATR project.

B.2 Requirements

This section describes many of the issues that need to be addressed or resolved before attempting to devise a complete design solution for the ATR Workbench. The constraints for this tool are both internal to the system and external from the requirement to integrate with other (yet to be designed) tools. These future external requirements create a lot of unknowns. However, the risk these unknowns create can be mitigated by requiring a system that is designed with general support for geographic and mapping functions, image processing, provides for rapid application development (scripting language, GUI builder), is flexible and easily integrates with other technologies (such as external libraries and code written in various languages). With this in mind, the requirements for the system are (in no particular order):

Internal - ATR Workbench:

- support for OMW, HNeT data (calling external batch processing)
- raster graphics support (full scene - zoom, pan, histogram)
- vector graphics support (user target center/vertical)
- platform independent
- appropriate for operational environment
- GUI support

External - Larger Toolkit:

- platform independent
- large raster image display
- vector support
- image processing
- geographic functions
- modules plug-in
- external library/function support
- research and operational environment
- rapid application development environment (GUI and scripting)

- fast processing (usually at odds with previous requirement)

B.3 Other Goals and Guidelines

The system must be designed using technology that will facilitate other tool development and integration into a combined system. As well, these tools contain a research element and therefore they should be quick and easy to develop, yet robust enough for testing operational use. The emphasis should therefore be on ease of development and integration of system components and not on speed or other resource type considerations. To ensure the long term and widespread adoption of the system among the group the technology should be an accepted industry standard with reliable support.

B.4 Evaluation

Four possible technologies are evaluated based on the above requirements.

B.4.1 Matlab

Matlab meets all the internal requirements. It supports displaying raster and vector graphics. It can run on any platform (with the purchase of a platform specific license). Code can be compiled for operational purposes, although with some difficulty (and another toolkit license). There is a built-in WYSIWYG GUI design tool (Guide).

As for the external tool requirements, to fit into a larger system of tools, Matlab meets most of the requirements. In addition to the above capabilities, it has support for image processing (Image Processing Toolkit) and geographic functions (Mapping Toolkit). External libraries can be supported through a C/Fortran/Java external API interface. The Matlab environment is well suited to rapid application development as it uses an interpreted scripting language and has a GUI building tool. The only requirements Matlab does not meet are the ability to handle large raster images, and the processing is generally not considered very fast (although functions can be implemented in C for speed improvements).

Another advantage of Matlab is that it is very popular, and is currently in wide use.

Overall, Matlab is an excellent platform and meets virtually all the requirements. The main disadvantages are cost (for toolkits, licenses), lack of large raster support and possibly lacking in some geographic functions and related file format support. Creating a system for operational testing using Matlab may also prove to be difficult.

B.4.2 IDL

IDL is very similar to Matlab in that it is a commercial development environment using an interpreted language.

It meets all the tool requirements except that WYSIWYG GUI building support is only available under Windows (however the resulting code build under Windows can be run on any platform). Platform independence is supported and licenses can be purchased for most popular platforms: UNIX platforms (SUN, DEC, IBM, SGI, and HP), Linux, Solaris x86 Windows 95/98/2000/NT, and Macintosh G3.

In terms of support for other systems IDL meets all the requirements. Large raster image processing is supported through a tiling mechanism. Geographic functions are well supported through the ENVI library. Applications can be compiled but still require the IDL runtime environment.

IDL seems to offer a very complete solution. Like Matlab, it also has the advantage that it is currently in wide use. In particular, the OMW (Ocean Monitoring Workstation) was written using it. OMW is also proof that a robust commercial application can be written using IDL. Further, since the ATR Workbench tool may be required to interface with OMW it may make integrating OMW into the system significantly easier. It may also be possible (depending on the OMW code licence) to extend OMW in place of developing an entirely new tool (based on different technology).

A disadvantage of IDL is that it does not appear to be a very widely used platform. This means there are fewer third-party libraries and support as seems to be the case with some of the other proposed solutions (such as Java and Matlab).

B.4.3 OpenEV (Python, Gtk and OpenEV)

First, a description of exactly what OpenEV is and the technologies involved. OpenEV is both a library and reference application for viewing and analyzing raster and vector geospatial data. Both the library and application are Open Source and licensed under the LGPL (you can sell applications that use the OpenEV library). The library is written in C and uses OpenGL for graphics rendering. C and OpenGL are very portable and versions of OpenEV exist for SGI, Sun, Linux and Windows.

Python is an object oriented scripting language that is also Open Source. It is interpreted and something of a cross between C++/Java/Perl. It is considered very human readable and most programmers easily pick it up in a day or two. The OpenEV library is designed to be used from Python. Python is available for most, if not all, popular platforms. Python code can also be byte-compiled to obscure the source code and speed up program execution. Many third-party libraries have been developed to provide an extensive set of (also free) tools for a variety of application domains.

The last component is The GIMP Toolkit (GTK) [44] which provides the GUI. Like the rest of the components it is also Open Source. It has both C and Python bindings and has been successful used under for SGI, Sun, Linux and Windows as well.

These combined components are currently used in Atlantis Scientific's latest commercial SAR and InSAR processors demonstrating that they can be successfully used in an opera-

tional product. Combined, these components meet all the requirements for the ATR tool. Additionally, a WYSIWYG GUI building tool, Glade, is available under Linux.

The main advantage of this solution is that it is completely free and we have complete access to the source code (since it is open source). OpenEV is designed specifically for geographic visualization applications and therefore has support for standard geographic file formats (through GDAL), mapping projections (through PROJ.4), vectors and display of large raster images (using a tiling techniques and OpenGL). The components are all platform independent and work well together to create an environment for both application development and research.

The main disadvantage of this solution is that finding software developers familiar with the technology is difficult and it may be difficult to get commercial technical support for these tools, should it be required. Since Atlantis Scientific developed OpenEV, they may be able to provide support for it. Additionally, there are one or two independent contractors who are experts in the software. Fortunately, there is a large body of user/community support for most of the OpenEV component technologies and therefore commercial support is not likely needed.

B.4.4 Java

Java is also a viable platform for development of the ATR tool. It is a widely used and supported cross platform language. Its large support has resulted in specialized image processing and geographic libraries that should facilitate development. There are also GUI classes and building tools that would permit easy development of a GUI. Java has been successfully used by ERSI in their ArcIMS and ArcExplorer (both free) products [45].

A quick search for image processing and display libraries revealed the following: Java 2D Library, Java Advanced Imaging Library, Java Imaging and Graphics Library (JIGL), Java Vision Toolkit (JVT). For geographic/mapping support there is GeoTools, a free Java toolkit. Building a GUI would likely use Java Foundation Classes (JFC) and Swing. A free GUI building tool called CockTail®was found.

Additionally, Java has support for large images by means of tiling.

Disadvantages of Java seem to be that its is does not facilitate quick application development. Java is expected to be better in this respect than C/C++, but it is still a compiled programming language. The advantage to this of course is that it is well suited for operational use as the many commercial applications prove. The main advantages of Java are platform independence, widespread use and support, and robust production quality nature. However, it is not designed as a research platform (as Matlab and IDL are) and the support for image processing and geographic/mapping functions is primarily found from third-party support.

Annex C
Pseudo-Classes

Throughout the ATR literature, it is common to define a class, typically called "Other" [27, 46, 47, 25, 48, 20] that contains all objects not belonging to any of the pre-defined classes upon which the classifiers are trained. Because membership in this Other class is well-defined, namely, it contains all elements in the complement of the union of the pre-defined classes,

$$C_O = \overline{\cup_j C_j}, \tag{C.1}$$

there is no difficulty in treating C_O like every other class.

In the ATR Workbench, there are two target type definitions employed that, for different reasons, do not have well-defined membership conditions and are therefore described as *pseudo-classes*. Additionally, there is an implicit third pseudo-class that fulfills the role of the Other class. The need for these pseudo-classes arises from the availability of multiple classifiers. Individual classification results may now be treated as intermediate evidence toward a cumulative decision, rather than as a definitive decision that would occur from single-classifier ATR. Thus, by designing the ATR Workbench to use these pseudo-classes, future implementations of multi-classifier decisions, evidential reasoning or classifier fusion may be better supported.

The Other class, as defined above, is partitioned into an explicit "Not Target" pseudo-class and an implicit "Other" pseudo-class, differentiating objects that the user wishes to retain in the target list from those to be outright rejected. Information about an image assigned the Not Target type will be maintained and further ATR analysis, such as classification using other ATR modules, is possible. Images that are deemed to obviously contain no target objects and not worth further analysis are discarded. Because different users may decide to place the same image in either the Not Target or Other pseudo-class, membership in each is subjective.

Rather than risk confusion by defining an Other type differently from (C.1), or by requiring an accurate understanding of the nuances and subtleties involved with distinguishing between the Other class and pseudo-class, assignment of this target type is treated implicitly in both the description and operation of the ATR Workbench.

Finally, the "Unknown" pseudo-class is unique in that its membership is known in advance to be the *empty set*. All objects will belong to one of the pre-defined, user defined or Other classes, and so one is able to assign with certainty a zero likelihood of membership in the Unknown pseudo-class to every image. Yet, there is good reason to choose not to do so. The assignment of Unknown type by an individual classifier provides a placeholder with zero weight for any cumulative decision made about an object. It is similar to the "No Decision" or "No Report" results used for time-sensitive classification [49]. For example, a reasonable approach would be that an image would not be assigned the Not Target type by automated methods until after every applicable ATR classifier yielded an Unknown result on it.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada - Ottawa
3701 Carling Avenue
Ottawa, ON K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Development of an ATR Workbench for SAR Imagery: Implementing a single user interface for automating target recognition processes (U)

4. AUTHORS (Last name, first name, middle initial)

English, Ryan A.; Rawlinson, Steve J. and Sandirasegaram, Nicholas M.

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
December 2002	43	49

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Defence R&D Canada - Ottawa
3701 Carling Avenue
Ottawa, ON K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)
5eu12	W7714-000464/001/SV

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)
DRDC Ottawa TR 2002-155	

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

In order to assist in the development of automation techniques for target recognition within SAR imagery, we have implemented an evaluation platform, the ATR Workbench. This will allow researchers and Geospatial Image Analysts (GIAs) to investigate the capabilities of various commercial and experimental applications, singly or in combination, as applied to the target recognition process. The platform will enable studies to determine which aspects improve GIA performance when automated, which methods best improve classifier performance, as well as which methods work better for particular environments and target class definitions.

The ATR Workbench was developed, using several open source tools, to provide a platform independent bridge between Automatic Target Detection (ATD) applications and target classifiers. It is capable of importing several kinds of ATD reports, of applying different feature extraction and preprocessing algorithms and of implementing various aspects of Automatic Target Recognition (ATR) applications while importing, displaying and reporting their results. Each step may be automated or operated interactively, as required. Initially, demonstrated upon the public MSTAR data set.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Automatic target recognition, ATR systems, assisted target recognition, SAR imagery exploitation, classification software, target segmentation, ATR evaluation

Defence R&D Canada

Canada's leader in defence
and national security R&D

R & D pour la défense Canada

Chef de file au Canada en R & D
pour la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

www.drdc-rddc.gc.ca