



Communications
Research Centre
Canada

An Agency of
Industry Canada

Centre de recherches
sur les communications
Canada

Un organisme
d'Industrie Canada

A Neural Network Classifier for ISAR Ship Imagery

Kenneth L. Sala
Communications Research Centre

Defence Research and Development Canada - Ottawa
3701 Carling Ave. Ottawa, ON Canada K1A 0Z4

Communications Research Centre Canada
3701 Carling Ave. Ottawa, ON Canada K2H 8S2

January 2006

TK5102.5
C673e
#2006-001
C.D.



Canada

CAUTION
This information is provided with the
express understanding that
proprietary and patent rights will
be protected





Communications
Research Centre
Canada
An Agency of
Industry Canada

Centre de recherches
sur les communications
Canada
Un organisme
d'Industrie Canada

A Neural Network Classifier for ISAR Ship Imagery

Kenneth L. Sala
Communications Research Centre

Defence Research and Development Canada - Ottawa
3701 Carling Ave. Ottawa, ON Canada K1A 0Z4

Communications Research Centre Canada
3701 Carling Ave. Ottawa, ON Canada K2H 8S2

January 2006

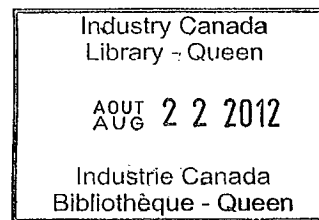
Canada

CAUTION
This information is provided with the
express understanding that
proprietary and patent rights will
be protected



A Neural Network Classifier for ISAR Ship Imagery


Kenneth L. Sala
Communications Research Centre



Defence Research and Development Canada - Ottawa
3701 Carling Ave.
Ottawa, ON Canada K1A 0Z4

Communications Research Centre
3701 Carling Ave.
Ottawa, ON Canada K2H 8S2

January 2006

- 
- © Her Majesty the Queen as represented by the Minister of National Defence, 2006
 - © Sa majesté la reine, représentée par le ministre de la Défense nationale, 2006

Abstract

An artificial neural network classifier is proposed and evaluated for the task of automated classification of ISAR ship imagery. Critical to the methodology employed in this research is the adoption of an image feature vector based exclusively upon the spatial Fourier spectrum of the ship image profile. This profile is derived from the original ISAR image following a sequence of image processing steps to filter, rotate, and enhance the ship image. The underlying motivation for the choice of a spatial Fourier spectrum to characterize an image and to serve as the input vector to the neural network classifier is the postulation that such a characterization combined with the learning capability of the neural network can effectively create an image classifier which encompasses and implicitly incorporates the classification criteria used by trained, human classifiers. The empirical results presented in this report strongly support this assertion in that perfect or near-perfect classification accuracies are routinely obtained for the cases where the neural network classifier is tested using images drawn from the same ISAR image folders used to train the network. In the case where the classifier is evaluated using images drawn from image folders not employed in the training phase, the classification accuracies are significantly lower but are found to quickly increase to acceptable levels as the number of distinct images folders used in the training stage is increased.

Résumé

Le présent document propose et évalue un classificateur de réseau neuronal artificiel servant au classement automatisé des images de navires réalisées à l'aide d'un radar à ouverture synthétique inverse (ROSI). La méthode employée par cette recherche exige l'adoption d'un vecteur de caractéristiques d'image fondé exclusivement sur le spectre spatial de Fourier du profil de l'image des navires. Ce profil est tiré de l'image initiale du ROSI ayant fait l'objet d'une série de traitements de l'image pour filtrer, tourner et améliorer l'image des navires. Le choix d'un spectre spatial de Fourier pour caractériser une image et servir de vecteur d'entrée du classificateur de réseau neuronal s'explique par le fait que la combinaison d'une telle caractérisation et de la capacité d'apprentissage du réseau neuronal peut efficacement créer un classificateur d'images qui englobe et intègre implicitement les critères de classification utilisés par des classificateurs humains formés. Les résultats empiriques présentés dans ce rapport semblent confirmer l'affirmation selon laquelle il est possible d'obtenir couramment un classement parfait ou presque parfait lorsque le classificateur de réseau neuronal est mis à l'essai à l'aide d'images provenant des mêmes dossiers d'images du ROSI utilisés pour former le réseau. Quand le classificateur évalue des images provenant de dossiers d'images inutilisés lors de l'étape de la formation, alors la précision du classement est beaucoup plus faible; toutefois, cette précision augmente à des niveaux acceptables à mesure qu'augmente le nombre de dossiers d'images distincts utilisés à l'étape de la formation.

Executive summary

An analysis is presented of an ISAR image recognition system which is based upon the use of an artificial neural network classifier and an image feature vector derived from the Fourier spatial frequency spectrum of the ship image profile. The empirical results confirm the viability and soundness of the premise that characterizing these types of ISAR images by a Fourier spatial frequency spectrum leads to accurate, reproducible image recognition using an artificial neural network classifier. The underlying assumptions that 'human classification rules' can be encompassed and enlarged upon by a Fourier spatial frequency spectrum and that an appropriately designed and trained neural network can discover the essential relationships in the spectral components needed to yield accurate pattern recognition are valid ones. The results from several empirical trials are given which examine the cases of testing the classifier on images drawn from folders used in the training sets versus testing the classifier on images drawn from folders not used in the training. When using images from common training-testing folders, essentially perfect classification results are found for as many as thirteen different ships. Conclusions are given along with suggestions for additional work in this area of image pattern recognition.

Sala, K.L. 2006. A Neural Network Classifier for ISAR Ship Imagery. DRDC Ottawa TR 2006-24/CRC RP-2006-001 Communications Research Centre.

Sommaire

Le présent document propose une analyse du système de reconnaissance des images de radar à ouverture synthétique inverse (ROSI) s'appuyant sur l'utilisation d'un classificateur de réseau neuronal artificiel et sur un vecteur de caractéristiques d'image tiré du spectre des fréquences spatiales de Fourier du profil de l'image des navires. Les résultats empiriques confirment la viabilité et la justesse du principe selon lequel la caractérisation de ces types d'images de ROSI par un spectre des fréquences spatiales de Fourier favorise la reconnaissance d'images reproductibles et précises à l'aide d'un classificateur de réseau neuronal artificiel. Deux hypothèses ont été validées : les « règles de classification humaine » peuvent être intégrées et élargies par un spectre des fréquences spatiales de Fourier, et un réseau neuronal adéquatement conçu et formé peut découvrir les relations essentielles des composants spatiaux requis pour obtenir une reconnaissance précise des modèles. Ce document compare les résultats de plusieurs essais empiriques concernant la classification des images provenant de dossiers utilisés dans les ensembles de formation ainsi que les résultats des essais portant sur la classification des images tirées de dossiers inutilisés lors de la formation. L'utilisation des images provenant de dossiers communs employés pendant la formation permet d'obtenir des résultats de classification presque parfaits pour un nombre maximal de 13 navires différents. Le document présente des conclusions ainsi que des suggestions visant à approfondir les travaux dans le domaine de la reconnaissance des modèles d'images.

Sala, K.L. 2006. A Neural Network Classifier for ISAR Ship Imagery. DRDC Ottawa TR 2006-24/CRC RP-2006-001 Communications Research Centre.

Table of contents

Abstract.....	i
Résumé	ii
Executive summary	iii
Sommaire.....	iv
Table of contents	v
List of figures	vi
List of tables	vii
1: Introduction	1
2: Preparation of the ISAR Image Dataset	4
3: Calculation of Feature Vectors for the ISAR Images	9
3.1 Overview	9
3.2 Image processing and feature vector calculation	9
3.3 Mensuration of the ship length	19
4: Evaluation of the Neural Network Classifier	26
4.1 Overview	26
4.2 The neural network classifier and training-test procedure	28
4.3 Empirical trials	30
5: Concluding Remarks	46
References	49
Appendix A	50

List of figures

Figure 2.1: Examples of rejected images.....	6
Figure 3.1: Header Information from Fingerprint Routine.....	12
Figure 3.2: (a) Raw ISAR image (b) High-intensity-pass filtered image.....	13
Figure 3.3: (a) Edge detected image (b) Radon transformation.....	14
Figure 3.4: (a) Rotated image (b) Vertical profile of rotated image.....	15
Figure 3.5: (a) Cropped, rotated image (b) Horizontal profile of rotated image.....	16
Figure 3.6: (a) Full FFT of horizontal profile (b) Feature vector of image.....	17
Figure 3.7: (a) Raw ISAR image (b) Bestwidth measurements.....	21
Figure 3.8: (a) Raw ISAR image (b) Bestwidth measurements.....	22
Figure 3.9: (a) Raw ISAR image (b) Bestwidth measurements.....	23
Figure 4.1: Improvement in classification accuracies with number of training folders.....	45

List of tables

Table 2.1: Final vetted image database.....	8
Table 3.1: Measurements of ship width for a single image folder.....	24
Table 4.1: Summary of Trial 1.....	31
Table 4.2: Summary of Trial 2.....	32
Table 4.3: Summary of Trial 3.....	33
Table 4.4: Summary of Trial 4.....	34
Table 4.5: Summary of Trial 5.....	35
Table 4.6: Summary of Trial 6.....	36
Table 4.7: Summary of Trial 7.....	37
Table 4.8: Summary of Trial 8.....	38
Table 4.9: Summary of Trial 9.....	39
Table 4.10: Classification Accuracies for Trial 3 using the RP Training Algorithm.....	41
Table 4.11: Classification Accuracies for Trial 3 using the CGB Training Algorithm.....	42

This page intentionally left blank.

1: Introduction

The task of automated image classification of SAR or ISAR imagery of military and commercial ships is a particularly challenging undertaking that involves simultaneously developing a rigorous, general means to characterize the underlying patterns in the individual images along with a pattern recognition methodology by which correlations in these patterns can yield accurate and robust image classification. Research in this area has been underway for several years and the Defense Research Establishment Ottawa has led much of the Department of Defense effort in this field with in-house and contracted projects. A variety of distinctly different approaches to this problem have been explored such as classical Bayesian classification, image mensuration analysis, template matching, and neural network pattern recognition.

In research efforts reported by Lockheed Martin Canada (Tessier and Shahbazian, 2000), a set of classification rules is presented which is derived from observations of how human classifiers arrive at a decision as to the category and/or identification of a particular ship or ship type. What is striking about this empirical set of rules is that the preponderance of these rules are based upon the recognition of concurrent spatial features within the image, i.e., a characteristic pattern is detected by the presence of two or more image features located at reproducible positions along the ship image profile (Gagnon L. and Klepko R., 1998, and Gouaillier V. and Gagnon L., 1997).

In the work presented in this report, the approach to defining an image pattern or feature vector to characterize individual images is based exclusively upon the spatial Fourier spectrum of the ship profile. It is felt that such a representation of the image embodies the underlying principle that the distribution of spatial Fourier components in the ship image profile is the optimal mechanism upon which to base an automated classifier. A neural network classifier is employed in this work (multi-layer perceptron network trained by backpropagation). It is a well established, empirical fact that a suitably designed neural network can serve, over a broad range of types of problems, as an accurate pattern recognition scheme and classifier provided that the quantitative measure

used to characterize the data is appropriately chosen (see Haykin 1994 and the Matlab manual for examples). In short, it is taken a priori that the methodology of neural network classification is fundamentally a sound one and that the principal task at hand is therefore the design and testing of the specific means for deriving a feature vector description of the ISAR images.

It is important to establish from the outset that the approach followed in the present work is primarily one of classification by means of individual ship identification and not one of classification of ships by category or type. The work reported by Lockheed Martin Canada in this area points to fundamental problems in attempting to categorize ships (military and commercial) either by class or category of ship. These difficulties arise largely because of the somewhat arbitrary, often subjective schemes adopted a priori to define, e.g., a particular ship as a destroyer escort as opposed to a destroyer. These classifications frequently involve operational intent or historical assignment of the particular ship and reflect neither subsequent refittings or upgrades nor recognize changing roles of other previously designated ships in the same or similar categories. Therefore, the work here adopts the precept that the classification of ISAR imagery of ships is carried out on a ship-by-ship basis and makes no attempt to recognize or classify ships as belonging to any pre-defined groups or categories. In other words, the approach adopted here treats each individual ship as defining a "class of one" so that there will exist as many classes as are ships within an image database. Classification of an unknown ship attempts therefore to find the closest match to a particular ship contained within the training set and, implicitly, recognizes the unknown ship as being in the same class or category of the matching ship. In certain cases, as would normally become apparent in the process of training the classifier, two (or more) distinct ships, for example, would be routinely misclassified one for the other. This would demand that a new class be defined characterized by imagery of both and either ship, i.e., the two (or more) ships would define a single class containing the two (or more) ships in question.

Chapter 2 of this report describes in some detail the ISAR image database provided by Defense Research and Development Canada in Ottawa and the various stages involved in the vetting of this set to

establish a working set of ship imagery consisting of data for 20 different ships. Chapter 3 discusses the specifics of the feature vector calculations for the images and the details of the neural network classifier and the training/testing procedures followed. The results of various trials employing different numbers of ships and combinations of training and test sets are given in chapter 4 along with a discussion of work carried out to provide a means of measuring ship image length. Results and conclusions drawn from the work presented in this report are summarized in chapter 5 along with a discussion of promising, specific topics of research which would motivate and guide future work in this area.

2: Preparation of the ISAR Image Dataset

The image dataset as originally received from DND consisted of 28,984 images contained in 220 folders. After removing empty folders, deleting corrupt, unreadable image files (as indicated by their file size), and deleting the unusable files for one particular target (for one ship, only 8 files existed and all 8 proved unusable), the remaining dataset consisted of

214 directories with 28,854 files

This set could then be logically divided into four groups:

/prospot_ctr	with 54 directories containing 6331 files
/prospot_roi	with 53 directories containing 6107 files
/prospot_ctr_H559	with 54 directories containing 8335 files
/prospot_roi_H559	with 53 directories containing 8081 files

The roi images were essentially identical to and hence redundant with the ctr images, differing primarily in the degree of contrast in the image, the result of a different ISAR processing methodology. It was decided to work with the ctr images exclusively since these images were judged overall to be superior in contrast and definition in general to their roi counterparts. The prospot_ctr set represented 19 different ships while the prospot_ctr_H559 represented a single ship only. Clearly, the distribution of images over the ships was highly skewed. While the H559 set comprised 57% of the unvetted total set, some ships were represented by as few as 4 images (0.03% of the unvetted set).

The very first task undertaken was to rename all of the image files. File names in the original set were duplicated in different folders. It was mandatory that each image file have a unique filename since, in the training and test sets ultimately formed for use with the neural network classifiers, the images files would be collected into common folders. This somewhat pedantic task was carried out using a MatLab routine that stepped through the various folders and renamed each image file with a name containing the flight, scene, and file numbers.

Following the creation of a duplicate image set with unique file names, a MatLab routine was written which read each image in the set and created both a positive and negative tiff image from the ISAR image. This was a pragmatic exercise which subsequently allowed the user to easily view and browse through any subset of the ship images using standard image viewing software (e.g., the browse function in Paint Shop Pro). This ability was frequently called on throughout all of the subsequent work with the image set.

A cursory examination of the image dataset indicated that a substantial portion of the original set represented images that were unusable for a variety of reasons. These included:

- (1) Obviously bad frame, mostly dark or white, no image present;
- (2) Multiple images within frame, i.e., no one clear ship image;
- (3) Image truncated, portion of ship image is missing;
- (4) Obvious 'smearing' of all or a portion of ship image vertically or horizontally;
- (5) Severe 'ghosting' of all or a portion of ship image; or
- (6) Ship image is very faint or portions of image not discernable

Examples of rejected images for each of the above reasons are shown in figure 2.1.

A MatLab routine was written which permitted the user, after selecting a particular folder, to view each image in that folder sequentially and to either accept or reject that image. Rejected images were moved to distinct folders containing rejected images only. In this way, the entire set was eventually split into two new image sets, i.e., the rejected images and the vetted (usable) images. In practice, the usual methodology was to run this routine twice on each folder. On the first pass, the obviously unusable images were rejected. The second pass through the folder then served to more critically (and, undeniably, somewhat subjectively) reject those images deemed unusable. The routine kept accumulatively a count of the total images rejected/accepted from the original folder. The final vetted sizes were

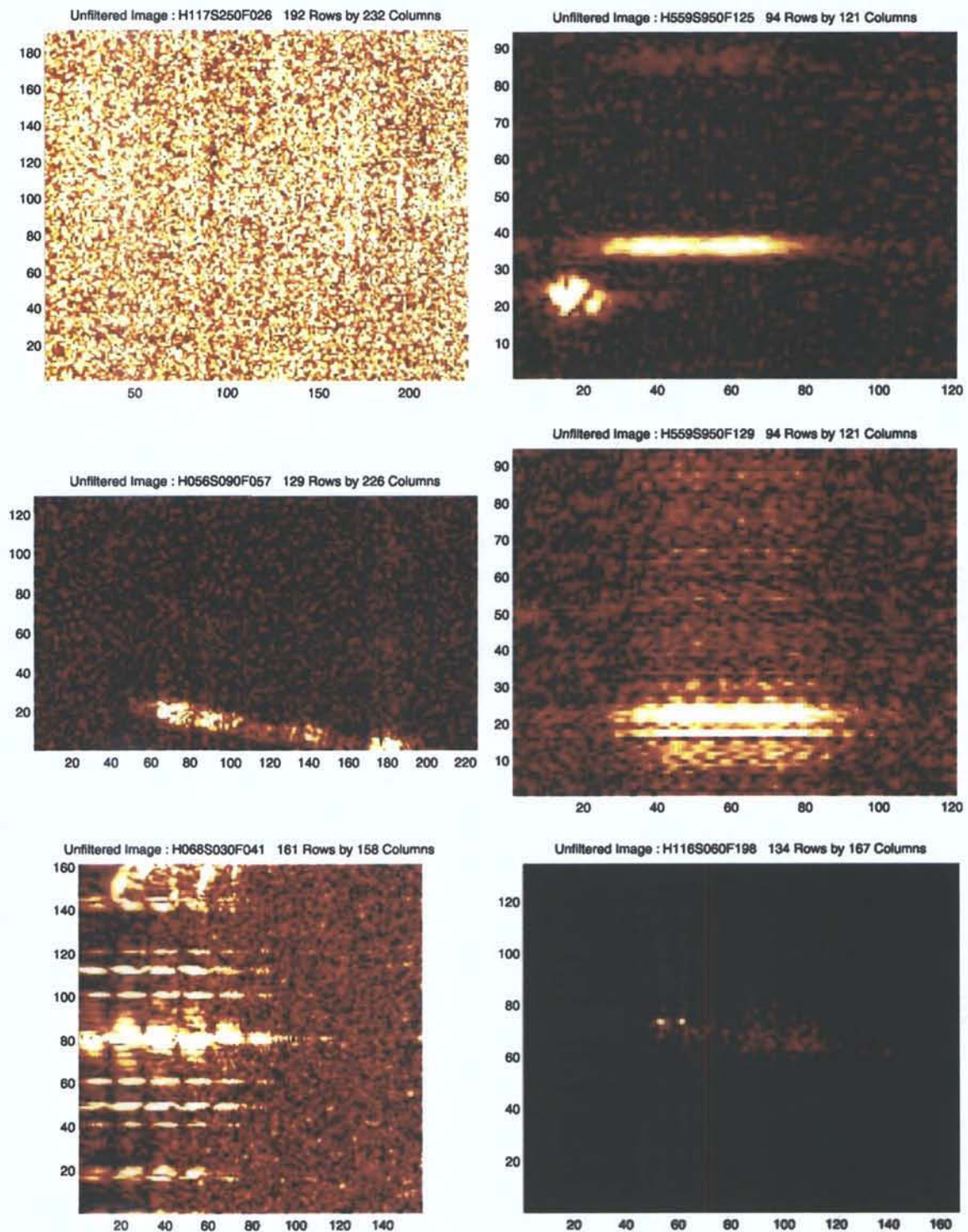


Figure 2.1: Examples of rejected images

/prospot_ctr_vetted with 54 directories containing 3826 files
/prospot_ctr_rejected with 54 directories containing 2505 files

revealing a 39.5% rejection rate. The corresponding results for the H559 set were

/prospot_ctr_H559_vetted with 54 directories containing 5546 files
/prospot_ctr_H559_rejected with 54 directories containing 2789 files

revealing a rejection rate of 33.5%. Thus the total vetted image set consisted of 9372 images (63.9% of the original 14666 images). The vetted image set was then grouped into 20 folders, each folder representing a single ship, and tiff images of this set were generated for reference purposes. Details of the final, vetted image data set are given in table 2.1. Note that several of the ships listed there are of limited usefulness in studies presented here since they are represented by a relatively small number of images contained in a single image folder. As well, it should be noted that the total number of images in the final image data set is considerably less than the number of 9372 stated above. This is chiefly the result of limiting the size of the dataset for ship number 20 to 12 folders and 1499 images, a pragmatic choice given the lesser number of folders and images employed for all other ships in the set.

CDROM	Original Directory Name	Directory Name Used	Ship No.	No. of Targets	No. of Images	Total
Prospot 2	h559s20_21_ctr	H559S200	20	1	151	1499
Prospot 2	h559s23_24_ctr	H559S230	20	1	233	1499
Prospot 3	h559s41_ctr	H559S410	20	1	108	1499
Prospot 3	h559s45_ctr	H559S450	20	1	121	1499
Prospot 3	h559s48_ctr	H559S480	20	1	123	1499
Prospot 3	h559s49_ctr	H559S490	20	1	111	1499
Prospot 3	h559s57_ctr	H559S570	20	1	105	1499
Prospot 3	h559s74_ctr	H559S740	20	1	119	1499
Prospot 3	h559s76_ctr	H559S760	20	1	102	1499
Prospot 3	h559s78_ctr	H559S780	20	1	101	1499
Prospot 4	h559s79_ctr	H559S790	20	1	102	1499
Prospot 4	h559s92_ctr	H559S920	20	1	123	1499
Prospot 1	h068s_30_0_ctr	H068S_30	1	1	30	617
Prospot 1	h068s_32_0_ctr	H068S_32	1	1	79	617
Prospot 1	h068s_51_0_ctr	H068S_50	1	1	75	617
Prospot 4	h270s40_ctr	H270S400	1	1	206	617
Prospot 1	h271s90_ctr	H271S900	1	1	104	617
Prospot 2	h475s14_ctr	H475S140	1	1	123	617
Prospot 1	h056s_13_0_ctr	H056S_10	2	1	74	519
Prospot 1	h056s_39_0_ctr	H056S_30	2	1	77	519
Prospot 1	h056s_9_0_ctr	H056S_90	2	1	74	519
Prospot 1	h468s34_ctr	H468S340	2	1	10	519
Prospot 2	h468s35_ctr	H468S350	2	1	68	519
Prospot 2	h468s51_ctr	H468S510	2	1	31	519
Prospot 2	h471s39_ctr	H471S390	2	1	89	519
Prospot 2	h471s42_ctr	H471S420	2	1	96	519
Prospot 1	h467s09_ctr	H467S090	3	1	165	408
Prospot 1	h467s17_ctr	H467S170	3	1	121	408
Prospot 2	h473s06_ctr	H473S060	3	1	122	408
Prospot 1	h116s_6_0_ctr	H116S_60	4	1	112	215
Prospot 1	h117s15_ctr	H117S150	4	1	103	215
Prospot 1	h056s_4_0_ctr	H056S_40	5	1	45	213
Prospot 1	h056s_6_0_ctr	H056S_60	5	1	92	213
Prospot 1	h115s_3_0_ctr	H115S_30	5	1	76	213
Prospot 1	h468s23_ctr	H468S230	6	1	120	196
Prospot 2	h471s18_ctr	H471S180	6	1	61	196
Prospot 2	h471s68_ctr	H471S680	6	1	15	196
Prospot 1	h068s_38_0_ctr	H068S_34	7	1	82	179
Prospot 1	h068s_43_0_ctr	H068S_40	7	1	46	179
Prospot 1	h068s_52_0_ctr	H068S_52	7	1	51	179
Prospot 2	h516s34_ctr	H516S340	8	1	90	172
Prospot 2	h516s36_ctr	H516S360	8	1	82	172
Prospot 1	h116s_5_0_ctr	H116S_50	9	1	73	143
Prospot 1	h117s25_ctr	H117S250	9	1	70	143
Prospot 1	h271s61_ctr	H271S610	10	1	103	135
Prospot 1	h271s62_ctr	H271S620	10	1	32	135
Prospot 2	h514s64_ctr	H514S640	11	1	9	132
Prospot 2	h514s68_ctr	H514S680	11	1	123	132
Prospot 2	h474s17_ctr	H474S170	12	1	120	120
Prospot 1	h038s_34_0_ctr	H038S_30	13	1	104	104
Prospot 1	h116s40_ctr	H116S400	14	1	95	95
Prospot 1	h067s_16_0_ctr	H067S_10	15	1	85	85
Prospot 2	h475s05_ctr	H475S050	16	1	55	55
Prospot 1	h067s_36_0_ctr	H067S_30	17	1	38	38
Prospot 1	h038s_55_0_ctr	H038S_50	18	1	31	31
Prospot 1	h058s_42_0_ctr	H058S_40	19	1	4	4

Table 2.1: Final vetted image database

3: Calculation of Feature Vectors for the ISAR Images

3.1 Overview

There exists a diversity of approaches to the task of classifying images which includes classical techniques such as statistical analysis, template matching, and principal component analysis, along with neural-network-based approaches based upon the concept of adaptive learning. As with any methodology for pattern recognition, the essential and arguably most important element in the design of a classifier is the choice for a quantitative measure to serve as, hopefully, a unique descriptor or 'fingerprint' by which the object may be classified. In the case of imagery, one rarely would use the raw image as such a measure since, (1), the amount of data and spurious detail could prove prohibitive and, (2), such data would be subject to wide variations owing to translations, rotations, and scaling of the image object. In practice, it is highly desirable to incorporate certain types of invariance (or, at least, a minimized sensitivity) in the image descriptor to variables such as scale and rotation of the object in the image plane.

In the present work, the descriptor or feature vector chosen to characterize an ISAR ship image is a normalized, spatial frequency of a one-dimensional profile of the image. Prior to calculating the spatial FFT, a series of processing steps are carried out to filter, normalize, rotate, and crop the image.

3.2 Image processing and feature vector calculation

A MatLab algorithm entitled "fingerprint", which is listed for reference purposes in Appendix A, was written to carry out the following sequence of tasks:

1. List the ship folders within a particular directory to allow the user to select (by number) a specific folder;
2. List the image files within the folder chosen in step 1 to allow the user to select (by number) one specific ship image;
3. Read the image file and header and normalize the intensity values to $\max = 1$ (all of the files have an initial $\max = 65535$);
4. Apply a high-pass intensity filter to the image to increase the contrast and reduce the amount of background noise;
5. Perform an edge detection operation on the filtered image;
6. Perform a Radon transform on the edge detected image;
7. Crop the Radon transform to exclude any detected lines close to $\pm 90^\circ$;
8. Determine the maximum value of the Radon transformation, rotate the image so that the principal line of the ship image is horizontal, and determine the new image dimensions of the rotated image;
9. Measure the vertical profile of the image (i.e., a row vector of the image integrated along the horizontal), determine the peak of the profile, and subsequently crop the rotated image to eliminate any vertical ghosting in the image and/or any spurious images;
10. Determine the (horizontal) profile of the filtered, rotated, cropped image;
11. Approximately center the ship profile and pad it to a dimension of 512 (for FFT purposes);
12. Calculate the FFT of the centered, padded ship profile;

13. Zero the DC component of the FFT;
14. Cut the FFT spectrum to a pre-determined length of specdim (= 64 throughout); and
15. Normalize the cut FFT spectrum to 1.

The resulting 1-D row vector of dimension 'specdim' with max = 1 constitutes the feature vector for the chosen image.

The 'fingerprint' routine described above examines a single, user chosen image file. This routine was readily amended to remove the prompts and to suppress the screen output so that it could be run recursively on a collection of images in a given folder. This generated a feature vector matrix for the training or test set of images and it was this matrix which was subsequently used in the training or testing of the neural network classifier, respectively. Figures 3.1 to 3.6 illustrate the results of the 'fingerprint' routine as it works through the various steps given above. Figure 3.1 shows the screen output from this routine listing the information read from the image file header.

A typical 'good' ISAR raw image is shown in figure 3.2(a) while the same image, after applying the high-intensity-pass filter, is shown in figure 3.2(b). The applied filter takes the form:

$$1 - e^{-\alpha I^2} \quad \text{where} \quad \alpha = -100 \ln(1 - OTL)$$

where OTL is the transmission level for $I = 0.10$ (max I for all images is = 1). The ad hoc value of $OTL = 10\%$ was used throughout this work.

The MatLab toolbox routine "edge" was invoked with the Sobel option to form an edge-detected version of the image, figure 3.3(a). This image was then used in another MatLab function "radon" to produce the Radon transformation shown in figure 3.3(b). The Radon transformation is used to detect the existence of linear structure in the original image in that the peak values found in the Radon transformation correspond to

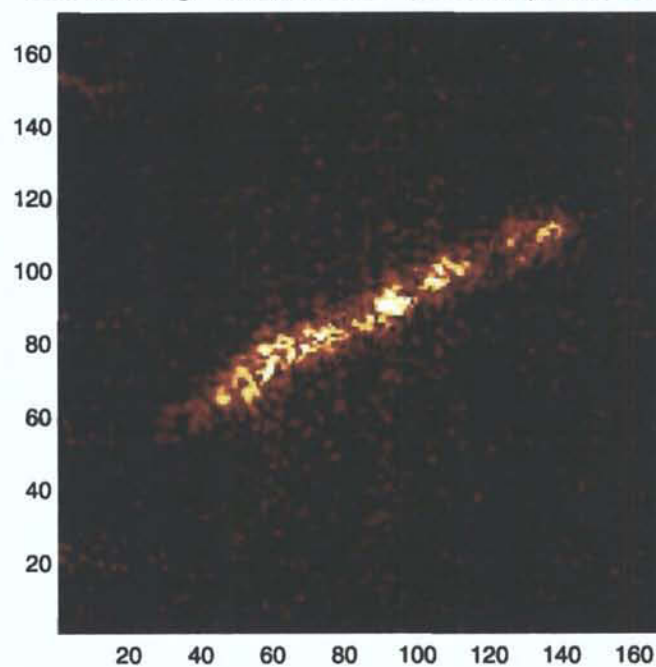
Processing Image d:\prospot\ship01\H068S030F026 168 Columns x 171 Rows

Image Header Information

pixelspacerrange	=	0.00	pixelspaceazimuth	=	0.00
pixelresrange	=	0.00	pixelresazimuth	=	0.00
imagefilever	=	1.00	imagedataformat	=	0.00
imageheadersize	=	2048	rangeframe	=	101039
sarmode	=	17.00	sartapevolno	=	68.00
targetselectmode	=	0.00	numberpulses	=	0
acheading	=	0.00	acaltitude	=	1.00
lfrate	=	52.73	samplingrate	=	0.00
squintangle	=	95.80	velocity	=	406.00
adaption	=	0	lfft	=	512.00
aperturetime	=	0.00	overlap	=	0.50
nramp	=	0.00	autofocus	=	1.00
rangecorrect	=	0.00	trackanglestart	=	91.40
xyaltitude	=	0.00	radialspeed	=	-8.44
frameheadersize	=	2048	firstsceneno	=	0
rangeofmax	=	178.00	azimuthofmax	=	269.00
latitude	=	1666972	longitude	=	4.292742e+009
scenenumber	=	255.00	lastframe	=	0
numbercolumns	=	168	numberrows	=	171
missionid	=	0 -18 -18 -18 0			
inversefilter	=	1469154535			
prf	=	200000000			
receivergain	=	1002159896			
rcmc	=	4.008636e+009			
timestamp	=	33554942			
maxpixelvalue	=	2.433650e-005			
FrameScaleFactor	=	2.692868e+009			
Product : MaxPixelValue x FrameScaleFactor	=	65535.00			

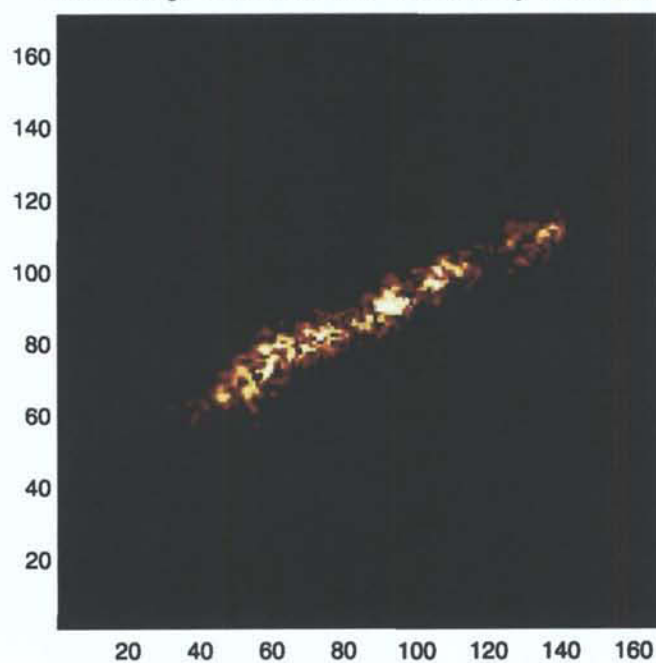
Figure 3.1: Header Information from Fingerprint Routine

Unfiltered Image : H068S030F026 171 Rows by 168 Columns



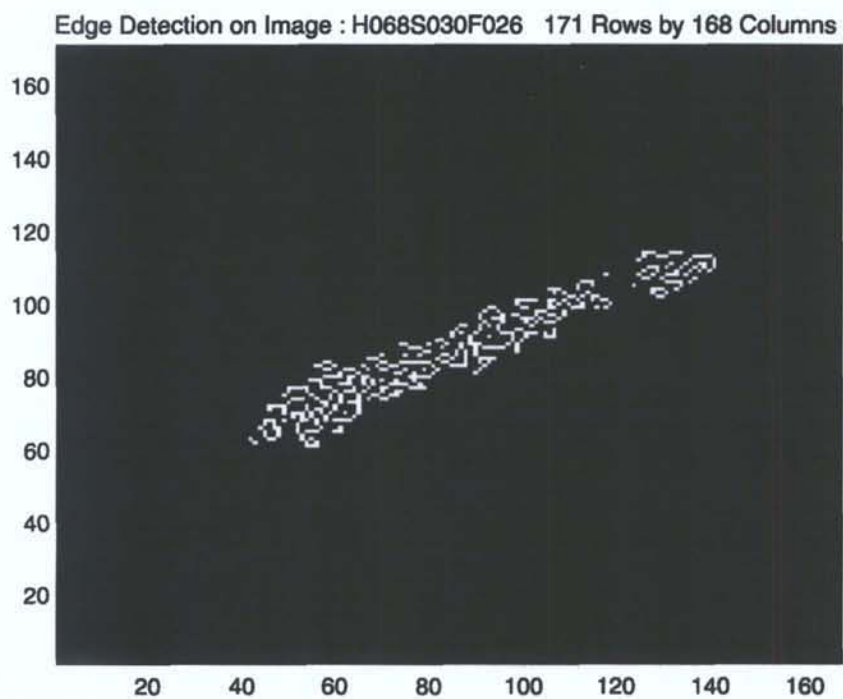
(a)

Filtered Image : H068S030F026 171 Rows by 168 Columns

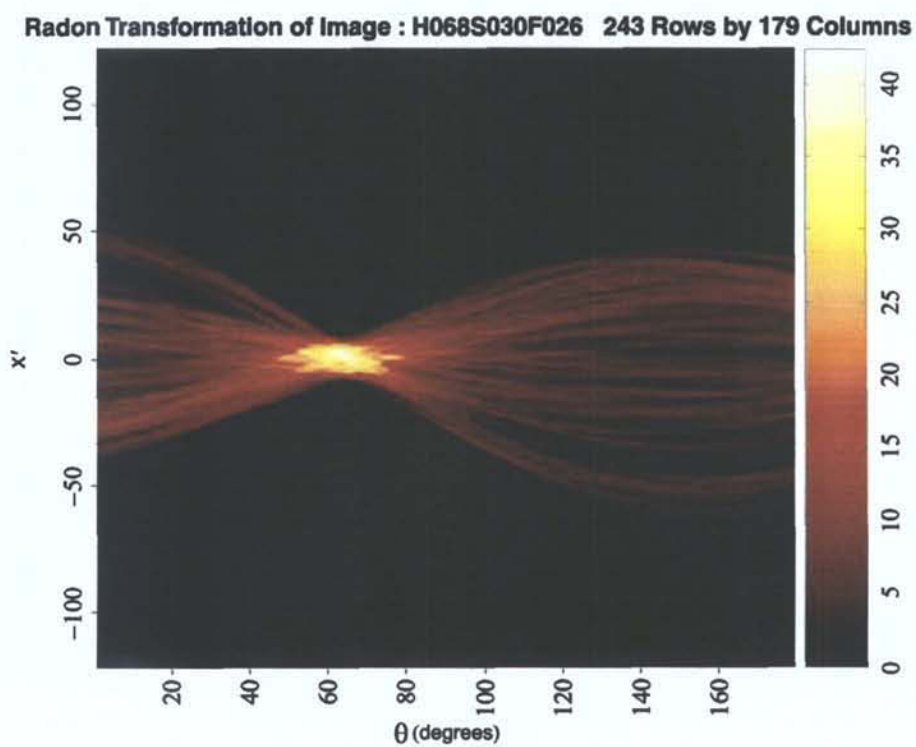


(b)

Figure 3.2: (a) Raw ISAR image (b) High-intensity-pass filtered image



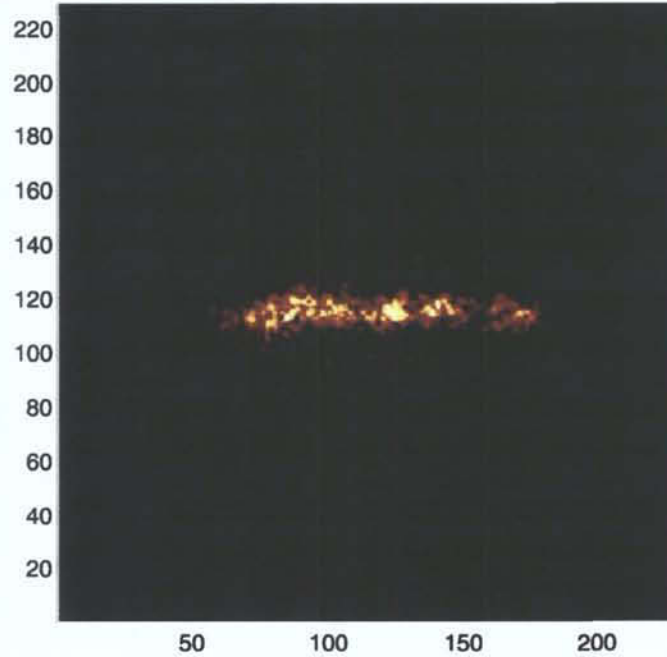
(a)



(b)

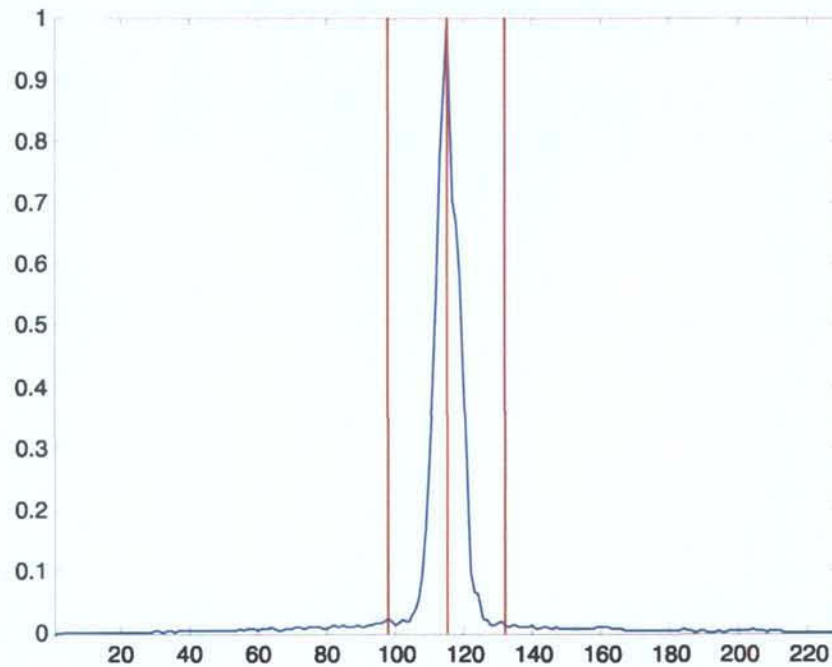
Figure 3.3: (a) Edge detected image (b) Radon transformation

Image : H068S030F026 Rotated -26 degrees 229 Rows by 229 Columns



(a)

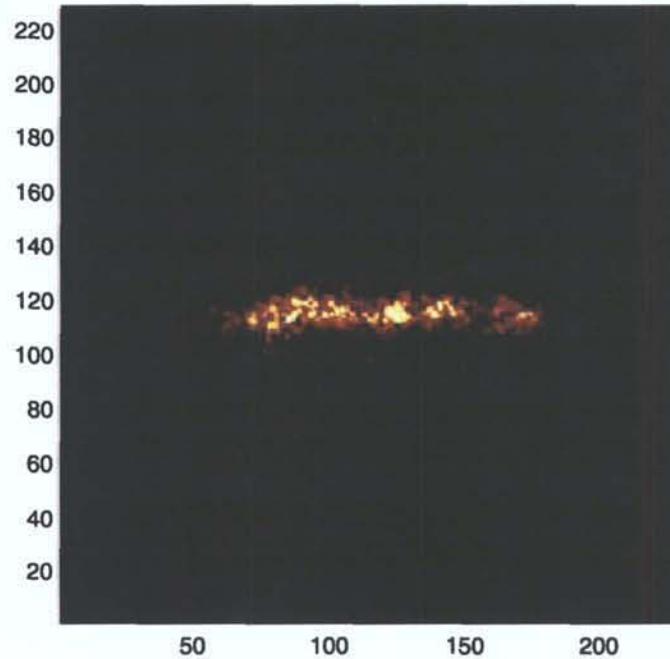
Vertical Profile with a Cutwidth = 34



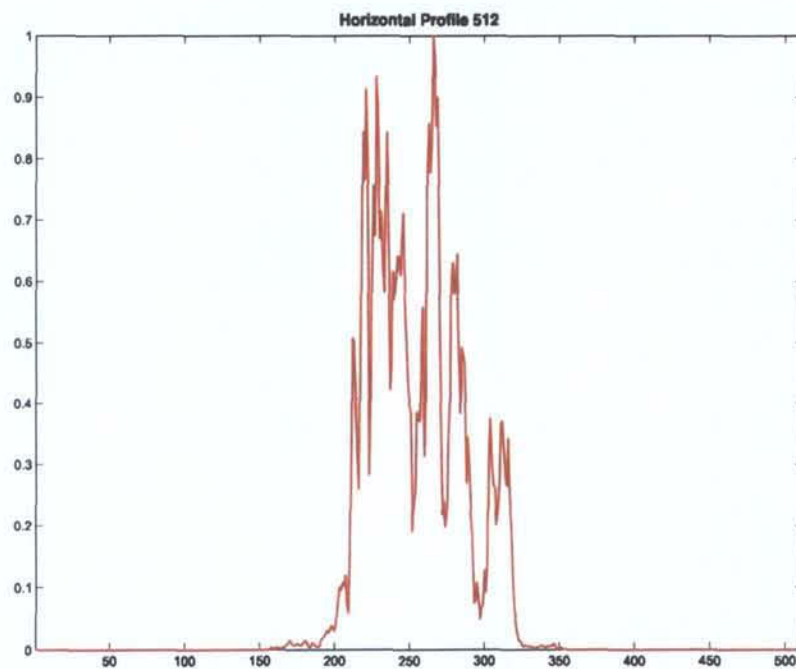
(b)

Figure 3.4: (a) Rotated image (b) Vertical profile of rotated image

Cropped Image : H068S030F026 Rotated -26 degrees 229 Rows by 229 Columns

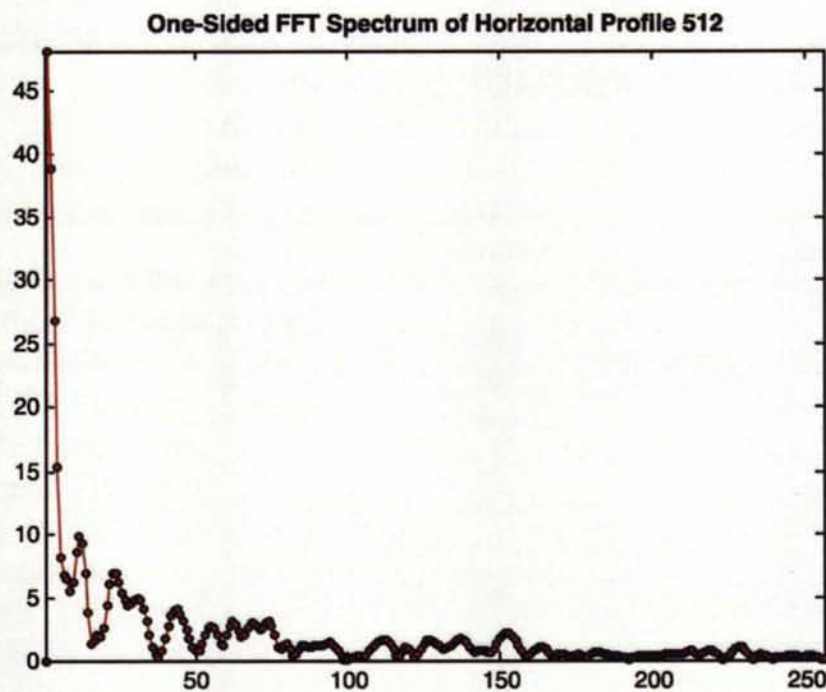


(a)

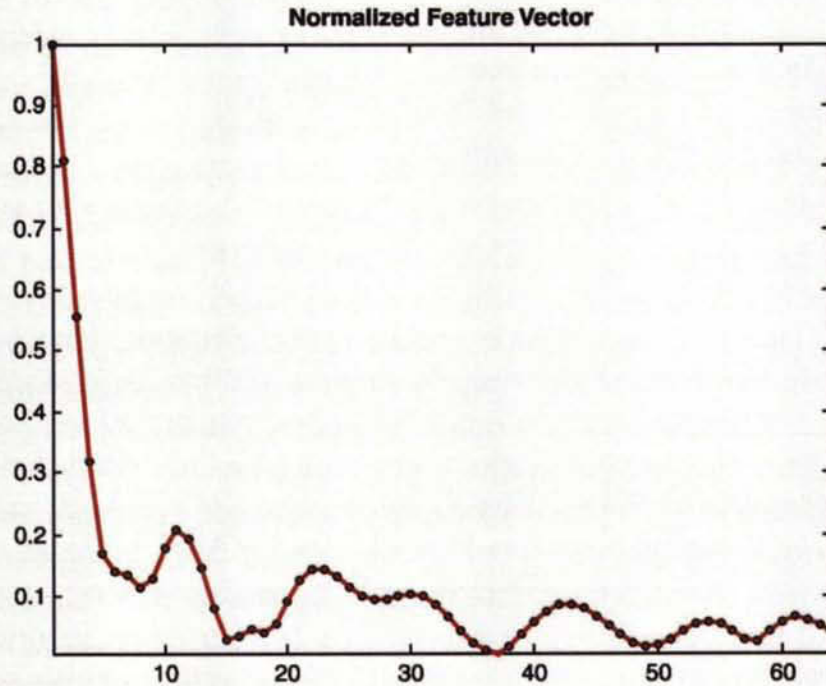


(b)

Figure 3.5: (a) Cropped, rotated image (b) Horizontal profile of rotated image



(a)



(b)

Figure 3.6: (a) Full FFT of horizontal profile (b) Feature vector of image

those angles at which 'line structure' is found in the image. In the example shown, the peak of the Radon transformation lies at a value of 64° which indicates a ship 'image axis' lying at $(64 - 90)$ or -26 degrees. The rotated image is shown in figure 3.4(a) with the axis of the ship image lying horizontal. It is noted that, very infrequently, the presence of a small, much fainter 'object' in the image (not part of the ship image per se) could produce a strong feature in the Radon transformation, particularly if the 'object' was a short, horizontal line. In some cases, this feature would be chosen by the fingerprint algorithm as that determining (incorrectly) the desired rotation of the image. In retrospect, such anomalies can be avoided completely by performing the Radon transformation not on the edge-detected image of figure 3.3(a) but instead directly on the filtered ship image of figure 3.2(b).

The image of figure 3.4(a) is then "scanned" along the vertical direction, i.e., the image matrix is summed along its rows to produce a vertical profile as in figure 3.4(b). This allows for the identification of the position of the image axis. The image is then cropped by zeroing all values above and below the cut lines (set at ± 15 pixels from the maximum). The resulting filtered, rotated, and cropped image, figure 3.5(a), is then "scanned" along the image axis, i.e., the image matrix is summed along its columns, to produce the ship profile, figure 3.5(b). It is this profile which now serves as the basis for application of the FFT to determine a spatial frequency spectrum.

The ship profile of figure 3.5(b) is "padded" to make its length exactly 512 pixels (the original profile is approximately centered in this padded profile – the FFT spectrum is invariant to the precise position of the original profile within the final profile) and an FFT is calculated for the ship profile. This yields a 256-point FFT shown in figure 3.6(a). It can be argued that the higher spatial frequencies in this spectrum are chiefly if not entirely determined by random noise and speckle in the original ISAR image and so the upper portion of the full FFT spectrum, along with the DC component, are discarded. The resulting portion of the FFT, normalized to one and running from the original frequencies from 2 to $(\text{specdim} + 1)$, is shown in figure 3.6(b). It is this spatial frequency

distribution that acts as the feature vector of the original image and serves as the input vector to the neural network classifier.

3.3 Mensuration of the ship length

As envisioned in the present work, the overall strategy for ship classification is to treat each individual ship as defining a distinct class. As the number of ships/classes is increased in the training database, it is anticipated that, at some point, a new ship added to this set will prove essentially indistinguishable from some existing ship already present in the imagery database (indistinguishable in the sense that both ships are misclassified as one another roughly 50% of the time). In such a case, a new "hybrid" class would be created composed of the two ships (the original classes corresponding to the individual ships removed from the training assignments) and subsequent testing of an unknown vessel could then compare to the class composed of the two similar ships.

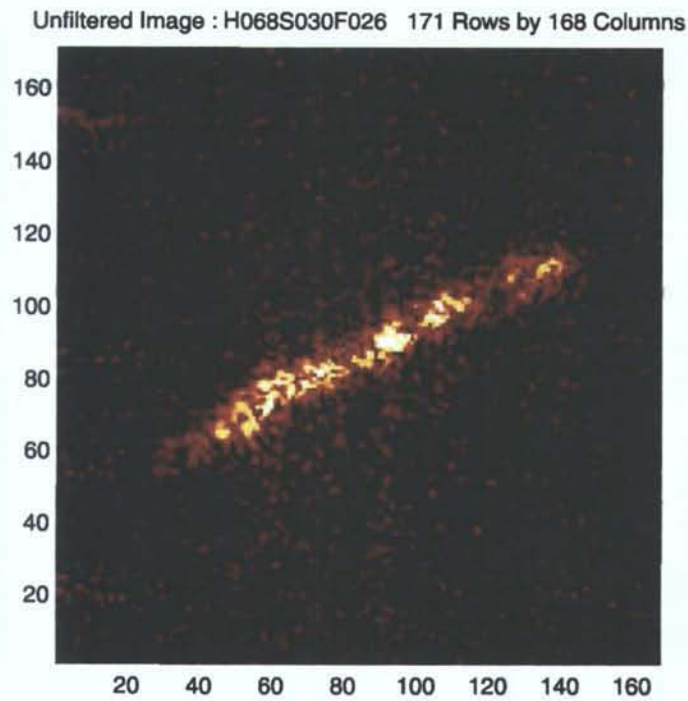
However, in such an approach, it is critical that the classifier be able to distinguish between disparate ships/classes which, although they may exhibit closely similar spatial frequency spectra, are nevertheless completely different vessels. By way of example, a small 12 meter gunship could, FFT-wise, appear indistinguishable from a much larger 120 meter frigate or destroyer. In such a case, it would be mandatory that the neural classifier be capable of clearly distinguishing between the two ships/classes. One of the obvious candidates for a feature vector component which could allow such classes to be separated would be a determination of the ship's absolute length. In the present work, since accurate ship-by-ship ground proofing was not available (and cannot be assumed to be present in any future testing and application of the classifier), the possibility of measuring the 'ship width' approximately by determining the 'width' of the filtered and rotated ship image was explored. It is stressed that, since the angle of viewing of the ISAR platform relative to the ship's orientation is an unknown, this approach to rotating the ship's image in the image plane is a heuristic one expected to yield an approximate but nevertheless reasonably accurate estimate of the

ship length. As well, it is required a priori that precise knowledge of the pixel resolution for range and azimuth directions be known.

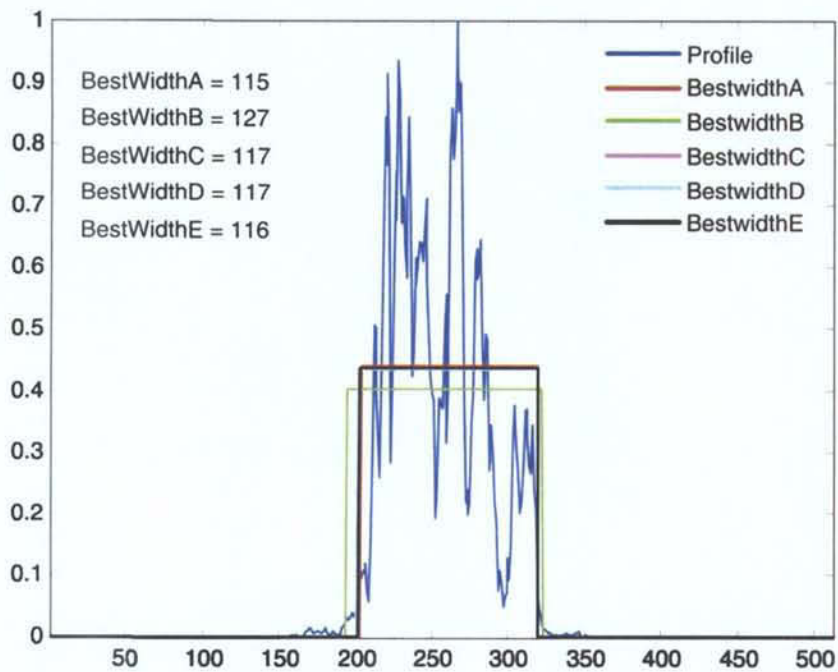
To this end, the 'fingerprint' MatLab routine incorporated five different approaches to such a mensuration task based upon the MatLab toolbox routines for calculating correlation and covariance vectors. Succinctly, the ship image profile (such as shown in fig. 3.5(b)) and a 'step width' square pulse (of a variable width from 1 to 512 pixels) together defined correlation and covariant vectors for the image. By examining the minmax properties of these vectors along with their points of discontinuity, estimates of the width of the image profile ("bestwidth") were calculated which attempted to allow for disparate cases of images partially or totally segmented owing to low image intensities and/or image corruption. The interested reader can find the details of these calculations in Appendix A.

For a 'clean' image such as that shown earlier in this chapter, all five approaches yield essentially identical results as illustrated in figure 3.7. Many images, however, tend to have low contrast and intensities resulting in 'dark gaps' within the overall image profile. These 'gaps' can then be misinterpreted by the mensuration calculations as illustrated in fig. 3.8 for the same ship (and folder) of the previous figure. In the particular case shown, only one of the five approaches yields the 'correct' result. Figure 3.9 illustrates a 'worst case' scenario for the type of image normally vetted and omitted from the final database. The image in fig. 3.9(a) shows a (false) second object in the image plane along with two 'ghosting' regions on either side of the ship image. The resulting bestwidth measurements shown in fig. 3.9(b) reveal four distinctly different width results with only one of the estimates being close to correct. It should be also noted that, when processed by the fingerprint routine, the image of fig. 3.9 is improperly rotated (by 0°), i.e., the image artifact on the left in fig. 3.9(a) and not the ship image becomes the basis for determining the image rotation. This error produces, in turn, an error (albeit a small one for the example shown) in the ship image profile.

Table 3.1 lists the results of the bestwidth measurements for 30 consecutive images drawn from a single image folder (the 'correct' width being approximately 110 pixels). This table reveals that no one approach



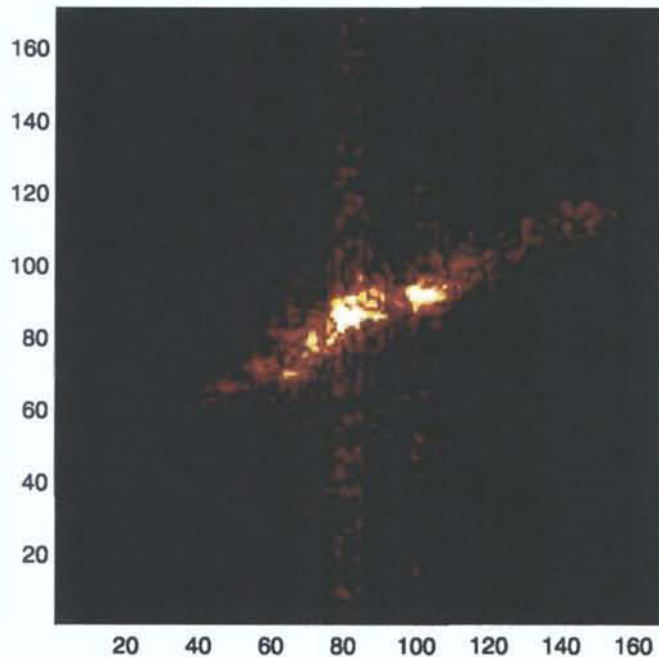
(a)



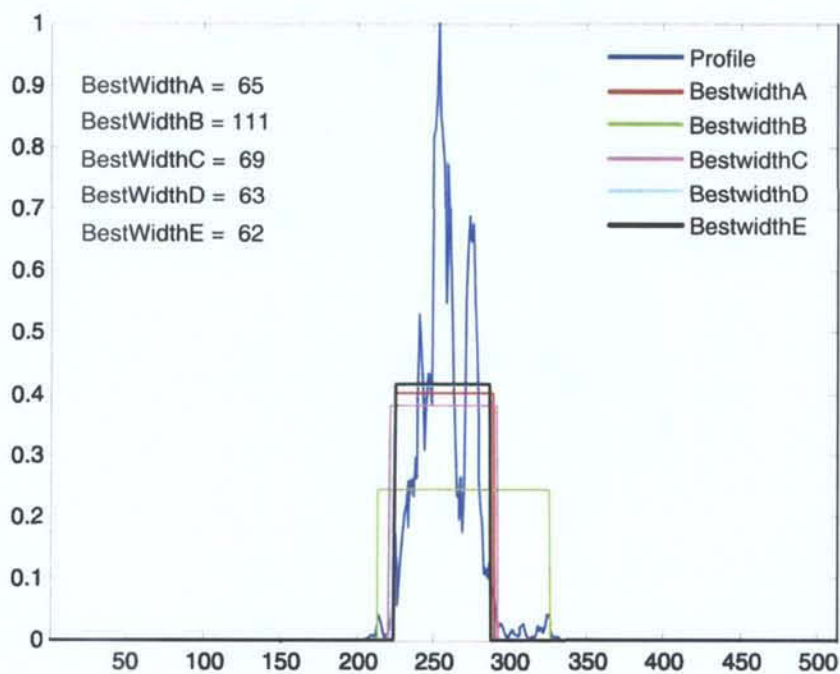
(b)

Figure 3.7: (a) Raw ISAR image (b) Bestwidth measurements

Unfiltered Image : H068S030F000 171 Rows by 168 Columns



(a)



(b)

Figure 3.8: (a) Raw ISAR image (b) Bestwidth measurements

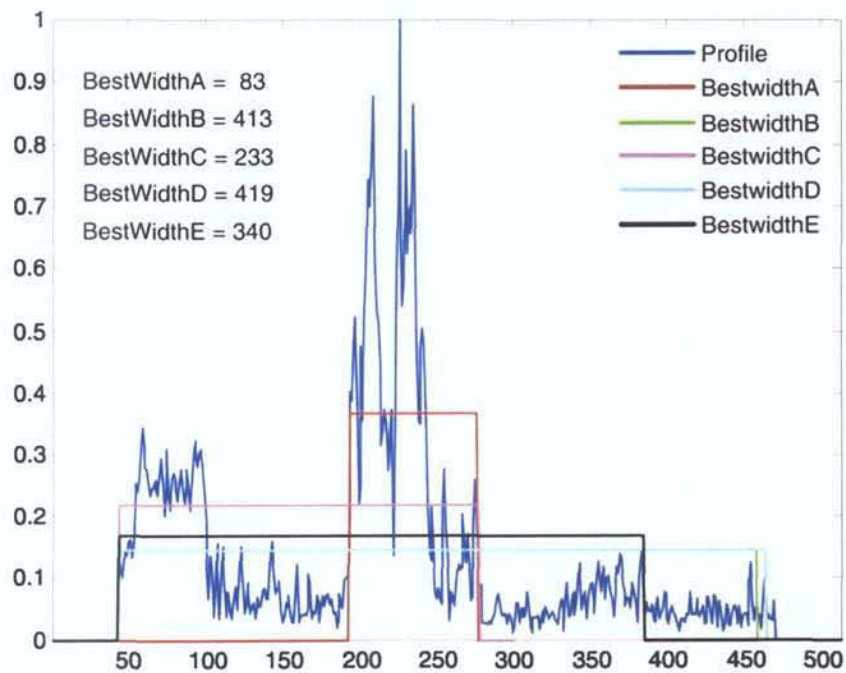
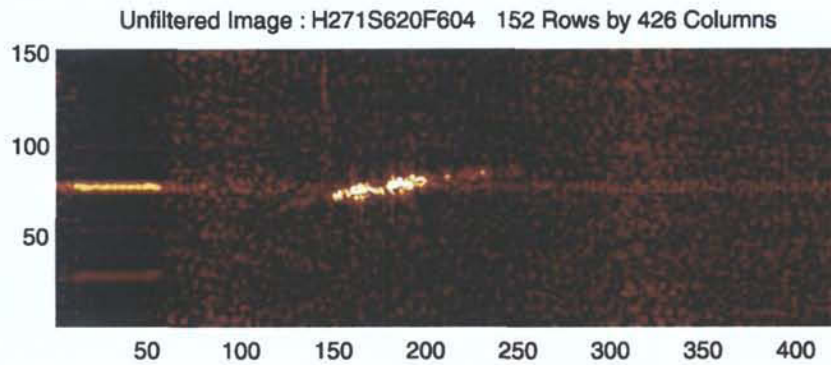


Figure 3.9: (a) Raw ISAR image (b) Bestwidth measurements

Ship Number	Filename	Bestwidth A	Bestwidth B	Bestwidth C	Bestwidth D	Bestwidth E	Rotation Angle
1	H068S030F000	65	111	69	63	62	28
1	H068S030F001	71	115	103	71	70	28
1	H068S030F002	65	105	97	91	60	19
1	H068S030F003	61	101	97	59	54	13
1	H068S030F004	61	95	93	55	54	6
1	H068S030F005	61	97	65	53	52	5
1	H068S030F006	67	99	73	59	58	4
1	H068S030F007	73	107	75	47	46	4
1	H068S030F008	101	105	103	99	92	8
1	H068S030F009	63	93	83	59	58	9
1	H068S030F010	63	103	105	93	62	11
1	H068S030F011	65	93	71	59	58	9
1	H068S030F012	67	93	69	67	66	10
1	H068S030F013	67	93	75	65	64	0
1	H068S030F014	65	99	77	63	62	0
1	H068S030F015	73	105	103	97	60	-8
1	H068S030F017	103	149	117	99	86	-15
1	H068S030F018	75	127	87	73	72	-7
1	H068S030F019	61	103	81	55	54	1
1	H068S030F021	61	97	73	71	60	-1
1	H068S030F025	81	143	117	111	110	34
1	H068S030F026	115	127	117	117	116	26
1	H068S030F027	97	107	101	97	96	22
1	H068S030F028	75	105	83	69	68	22
1	H068S030F032	77	93	77	69	62	-6
1	H068S030F043	69	101	79	67	66	-5
1	H068S030F044	63	127	67	109	108	16
1	H068S030F045	55	67	61	53	52	17
1	H068S030F046	59	105	69	59	50	10
1	H068S030F053	79	107	107	103	102	-2

Table 3.1: Measurements of ship width for a single image folder

is entirely accurate in all cases and that it is possible for all of these different techniques to fail for particular images. Such a finding would come as no surprise to any reader familiar with the fundamental difficulties encountered in image mensuration algorithms in general. The data in figure 3.10, as will be discussed in the conclusions, suggests that a better approach to the task at hand would be to utilize the image sequence characteristics of SAR and ISAR imagery to more 'intelligently' arrive at estimates of physical parameters such as ship image width by employing weighted averages of such measurements taken over similar images from within an image folder.

4: Evaluation of the Neural Network Classifier

4.1 Overview

The primary aim in the design and testing of a neural network classifier for ISAR ship imagery is to evaluate the soundness of the underlying methodology of characterizing the imagery by the use of a Fourier spatial spectrum and not the assessment or optimization of the neural network classifier by empirical testing of its operational parameters. In other words, it can be assumed a priori that a suitable neural network paradigm (such as a multi-layer perceptron network trained by a backpropagation algorithm) can accurately and robustly classify such imagery provided that the feature vector employed to characterize an image serves as an appropriate means by which images can be distinguished from each other. Thus, the intent of the testing trials conducted for the ISAR imagery is to determine the legitimacy and applicability of the Fourier spatial frequency characterization of the imagery as a sound feature vector representation of the ship images from an information-theoretic point of view. To this end, extensive investigation of the neural network classifier was not carried out in the sense that no systematic study was undertaken of the neural network architecture, the various types of distinctly different training methodologies, the effects of neural transfer functions, feature vector normalization, or of any of several other neural network design features or parameters which can affect a classifier's performance beyond those simple variations or choices for such parameters necessary to ensure convergent, reproducible training.

As the results to follow will clearly indicate, it is necessary to fundamentally draw a distinction between images drawn from a single data folder (same flight and scene number) and images drawn from different folders, whether or not the latter are for the same or different ships. There is a strong degree of correlation among the images within a single folder. The degree of correlation between same-ship folders is considerably less so that the success of any classification scheme will

ultimately rest, in large part, on the question of the degree of correlation between same-ship folders being stronger than any 'residual' degree of correlation between folders representing different ships. The results presented in this section will therefore be distinguished unambiguously according to whether or not the images used in the test set are drawn from the same ship folders from which the training set images were drawn or drawn from folders whose images were not used in forming the training set. The most realistic scenario and the one which provides the more stringent test of a classifier is that in which the test images are taken from folders not seen by the neural network in the training phase.

The final point to be clarified concerns the adoption in this work of the "one ship = one class" concept as opposed to any attempt to a priori define ships as belonging to distinct naval categories such as destroyers, frigates, escorts, etc. or to non-military categories such as commercial freighters, oil tankers, and so on. As detailed in the Tessier and Shahbazian report, the categorization of military ships into broad groups such as destroyers and frigates is a largely artificial exercise which leads to, in many cases, obvious discrepancies and inconsistencies when assigning a particular ship to one pre-defined category. Factors such as legacy assignments, refitting and upgrading of a ship's structure and armaments, and inherent variability of ship size, speed, etc. within any category make it pragmatically unattainable for a classifier (neural network or otherwise) to provide consistent and accurate assignment of ships to such broadly defined and variable categories. By way of just one example, many modern destroyer escorts are larger, faster, and more heavily armored than older ships originally commissioned as destroyers. The approach followed here is to treat each ship as defining a single class, presumably distinguishable from all other ships/classes by the appropriate classifier. Examination therefore of an unknown ship would lead to its classification as being most similar to a known ship used in the classifier training process. In practice, it could be anticipated that, as the number of ships used in the training phase is increased, certain pairs of ships would prove to be effectively redundant or equal from the classifier's point of view. In such cases, new classes would be defined which would, in effect, be a (possibly weighted) collection consisting of images drawn from these "indistinguishable" matches.

4.2 The neural network classifier and training-test procedure

The basic neural network architecture used throughout these studies was a three-layer feedforward perceptron network trained using four different variations of the backpropagation training algorithm. The first layer of the network is a distribution layer accepting all of the feature vector components at each node and forwarding a weighted sum of these components to the succeeding, 'hidden' layer. The outputs from the 'hidden' layer are then forwarded to the third, output layer which produces values between 0 and 1 at each node. In all cases, the network dimensions were [64 x 32 x numclasses] where numclasses is the number of classes (not necessarily the number of ships) defined for the training/test sets. The MatLab Neural Network Toolbox was used exclusively to define, monitor, train, and test the neural classifier. Logarithmic-sigmoid transfer functions were employed for both the input-to-hidden layer and the hidden-to-output layer. The neural classifier was an analog device with the output values in the range of [0,1].

The four different variations of the training algorithm used were:

RP - Resilient Backpropagation
SCG - Scaled Conjugate Gradient
GDX - Variable Learning Rate Backpropagation
CGB - Conjugate Gradient with Powell-Beale Restarts

Other training algorithms were tested but rejected because of a failure to converge, consistently poor classification results, and/or excessively long training times. Of the four algorithms tested, the RP and SCG routines generally provided the fastest and most consistent training results.

Each trial consisted of three distinct steps:

1. Preparation of separate training and test image sets.
Images were selected at random from the source image folders to compose the training set. The test set images then either consisted of the remainder of the images from these folders or images drawn from different folders not used in the training set.

2. Generation of the training and test feature vector matrices.
An automated version of the MatLab routine 'fingerprint' was written which calculated the feature vectors for the entire set of training and test images by iteratively stepping through each image in an assigned data folder. The individual feature vectors were normalized to 1. The final result was a matrix of dimension [numtrain x 64] or [numtest x 64] where numtrain and numtest are the dimensions of the training and test sets, respectively. These matrices served as the direct inputs to the neural network classifier.
3. Training and testing of the neural network.
A relatively short and simple MatLab routine was written which, after first defining the neural network architecture and basic operating parameters, trained the classifier and then calculated the classification matrix for the assigned test set. This routine was iteratively run 100 times with a random initialization of the untrained network at the commencement of each run. The network and all the result matrices were saved as mat files and a "winner" was chosen from the 100 runs as that run producing the smallest least-mean-square error for the test set. A separate MatLab routine was written which read these result mat files and carried out an analysis of the network classifier's performance. Training times varied considerably from trial to trial and for the different training algorithms with times as short as minutes (for the algorithms RP and SCG on the smallest, single folder training sets) and as long as days (30 or more hours) for the GDX and CGB algorithms on the trials involving large numbers of classes and/or large numbers of training images.

The training of the neural network classifier proceeded iteratively, stepping through each image in the training set with the order of the images randomized upon each pass through the set. The modification of the neural network weights was carried out by the MatLab training routine either upon completing a calculation for the entire training set,

i.e., one epoch, or, depending upon the specific training algorithm, after the result for a subset of the training set was calculated, the size of the subset determined by the dimensions of the training set. The training of the network ended upon reaching one of the three following criteria:

1. Network performance (the calculated error for the test set) reached a predefined lower limit. This limit was set to 10^{-7} for all trials.
2. The error gradient of network performance fell below a predefined lower limit, indicating that no further training was occurring. This limit was set to 10^{-8} for all trials.
3. The maximum number of epochs (training passes through the entire training set) was reached without reaching either of the limits for network error or error gradient. The maximum number of training epochs was set to 4000 for all trials (a small number of training/test runs were conducted with a higher limit for trial 3 as described in the notes for this trial in the trial summaries to follow).

When conducting the trials using multiple runs, the MatLab routine calculated and saved a record of the number of times the training process ended according to the three criteria above. This record provided valuable insight to how well any given training algorithm dealt with the particular training set under study.

4.3 Empirical trials

In all, nine complete trials were conducted using all four training algorithms for a total of 36 measurements of network performance. The results of these trials are given in the trial summary sheets of Table 4.1 to 4.9. The trials consisted of three related groups, namely trials 1 to 3, trials 4 and 5, and trials 6 to 9. Trials 1 to 5 employed “common” folders, i.e., test images were drawn from the same folder(s) used to compose the training sets, while trials 6 through 9 employed “exclusive” folders, i.e., test images were drawn from folders which were not used in the training

Trial Number = 1

Number of Ships = 3

Number of Classes = 3

Training Set = 300 images

Test Set = 304 images

Training/Test Image Folders = Common

Trial Description = 3 Classes, 3 Ships, Single Folder per Class

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S230	20	100	133
H270S400	1	100	106
H467S090	3	100	65
Totals		300	304

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 20	Ship 1	Ship 3	
RP	100	99	1	0	100.0%	100.0%	100.0%	100.0%
SCG	100	98	2	0	100.0%	100.0%	100.0%	100.0%
GDX	100	100	0	0	100.0%	100.0%	100.0%	100.0%
CGB	100	73	29	0	100.0%	100.0%	100.0%	100.0%

Notes: Perfect or near perfect classification for all 100 runs with training algorithm RP and GDX
 Perfect or near perfect classification for 98 runs with training algorithm SCG. 2 runs producing a result of 100,0,0
 "Hit or miss" results with training algorithm CGB. Many runs perfect or near perfect but many with 1 or 2 classes 100% misclassified

Table 4.1: Summary of Trial 1

Trial Number = 2

Number of Ships = 3

Number of Classes = 3

Training Set = 300 images

Test Set = 304 images

Training/Test Image Folders = Common

Trial Description = 3 Classes, 3 Ships, Single Folder per Class

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number		Training Images	Test Images
H471S420	2		70	26
H116S060	4		70	42
H468S230	6		70	50
Totals			210	118

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 2	Ship 4	Ship 6	
RP	100	100	0	0	100.0%	100.0%	100.0%	100.0%
SCG	100	100	0	0	100.0%	100.0%	100.0%	100.0%
GDX	100	100	0	0	100.0%	100.0%	100.0%	100.0%
CGB	100	74	26	0	100.0%	100.0%	100.0%	100.0%

Notes: A repeat of trial 1 with three different ships
 Perfect classification for all 100 runs with training algorithm RP, SCG, and GDX
 "Hit or miss" results with training algorithm CGB. Majority of runs perfect or near perfect but several with 1 or 2 classes 100% misclassified

Table 4.2: Summary of Trial 2

Trial Number = 3

Number of Ships = 13

Number of Classes = 13

Training Set = 910 images

Test Set = 532 images

Training/Test Image Folders = Common

Trial Description = 13 Classes, 13 Ships, Single Folder per Class

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S200	20	70	81
H475S140	1	70	53
H471S420	2	70	26
H473S060	3	70	52
H117S160	4	70	33
H056S060	5	70	22
H469S230	6	70	50
H516S340	8	70	20
H271S610	10	70	33
H514S680	11	70	53
H474S170	12	70	50
H036S030	13	70	34
H116S400	14	70	25
Totals		910	532

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result													Average Class-ification Accuracy
					Ship 20	Ship 1	Ship 2	Ship 3	Ship 4	Ship 5	Ship 6	Ship 8	Ship 10	Ship 11	Ship 12	Ship 13	Ship 14	
RP	100	14	86	0	93.8%	100.0%	100.0%	100.0%	100.00%	100.00%	96.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.90%
SCG	100	71	29	0	95.1%	100.0%	100.0%	100.0%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.20%
GDX	100	0	0	100	95.1%	100.0%	100.0%	100.0%	100.00%	100.00%	96.00%	100.00%	97.00%	100.00%	100.00%	100.00%	100.00%	98.90%
CGB	100	27	73	0	95.1%	100.0%	100.0%	100.0%	97.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.10%

Notes: Perfect or near perfect classification for all 100 runs with training algorithm RP.

Perfect or near perfect classification for most runs with training algorithm SCG. Many runs producing a result with from 1 to 10 classes 100% misclassified

Results with training algorithm GDX similar to that for SCG. CGB also "hit and miss" but with many more runs exhibiting 100% misclassifications

20 additional runs were carried out for the GDX algorithm with max. no. of epochs increased to 10,000 with no measurable improvement in results

Table 4.3: Summary of Trial 3

Trial Number = 4

Number of Ships = 3

Number of Classes = 9

Training Set = 720 images

Test Set = 600 images

Training/Test Image Folders = Common

Trial Description = 9 Classes, 3 Ships, Each Folder a Class

Mean Squared Error Goal = 10^{-7} Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S230	20	80	153
H559S480	20	80	43
H559S920	20	80	43
H270S403	1	80	126
H271S903	1	80	24
H475S143	1	80	43
H467S090	3	80	85
H467S170	3	80	41
H473S063	3	80	42

Totals 720 600

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Ship 20	Ship 20	Ship 20	Ship 1	Best Result Ship 1	Ship 1	Ship 3	Ship 3	Ship 3	Average Classification Accuracy
RP	100	19	81	0	96.1%	97.7%	100.0%	96.8%	100.0%	97.7%	97.6%	100.0%	97.6%	97.5%
SCG	100	68	32	0	98.7%	100.0%	100.0%	99.2%	100.0%	100.0%	100.0%	100.0%	100.0%	99.5%
GDX	100	0	0	100	98.7%	100.0%	100.0%	99.2%	100.0%	100.0%	100.0%	100.0%	100.0%	99.5%
CGB	100	41	59	0	98.7%	100.0%	100.0%	99.2%	100.0%	100.0%	100.0%	100.0%	100.0%	99.5%

Notes: This trial combined with trial 5 tests the intrinsic distinguishability between same-ship folders as opposed to the distinguishability of ships represented by multiple folders
 Both the RP and GDX algorithms produced near perfect classifications with some 'hit and miss' results for the GDX method
 The SCG and CGB algorithms produced similar results to the RP and GDX algorithms but with considerably greater number of results exhibiting one or more 100% misclassified classes

Table 4.4: Summary of Trial 4

Trial Number = 5

Number of Ships = 3

Number of Classes = 3

Training Set = 720 images

Test Set = 600 images

Training/Test Image Folders = Common

Trial Description = 3 Classes, 3 Ships, Three Folders per Class

Mean Squared Error Goal = 10^{-7} Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S230	20	80	153
H559S480	20	80	43
H559S920	20	80	43
H270S400	1	80	126
H271S900	1	80	24
H475S140	1	80	43
H467S090	3	80	85
H467S170	3	80	41
H473S060	3	80	42

Totals

720

600

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 20	Ship 1	Ship 3	
RP	100	1	99	0	100.0%	100.0%	100.0%	100.0%
SCG	100	0	100	0	100.0%	100.0%	100.0%	100.0%
GDX	100	0	100	0	100.0%	100.0%	100.0%	100.0%
CGB	100	3	97	0	100.0%	100.0%	100.0%	100.0%

Notes:

This trial combined with trial 4 tests the intrinsic distinguishability between same-ship folders as opposed to the distinguishability of ships represented by multiple folders

All 4 training algorithms produced perfect or near perfect classifications for all (RP and GDX) or most (SCG and CGB) runs.

The SCG and CGB algorithms produced results with 1 or 2 100% misclassified classes, a small number for the SCG method and considerably more (~ 25%) for the CGB algorithm

Table 4.5: Summary of Trial 5

Trial Number = 6

Number of Ships = 3

Number of Classes = 3

Training Set = 180 images

Test Set = 425 images

Training/Test Image Folders = Exclusive

Trial Description = 3 Classes, 3 Ships, 1 Train Folder per Ship

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S200	20	60	0
H559S480	20	0	123
H475S140	1	60	0
H270S400	1	0	206
H058S030	2	60	0
H472S420	2	0	96
Totals		180	425

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 20	Ship 1	Ship 2	
RP	100	52	48	0	91.1%	53.4%	83.3%	71.1%
SCG	100	46	54	0	86.2%	57.8%	87.5%	72.7%
GDX	100	70	30	0	87.8%	68.9%	60.4%	72.5%
CGB	100	34	66	0	87.0%	64.6%	82.3%	75.1%

Notes:

Trials 6, 7, 8, and 9 measure the improvement in classification accuracies as the number of folders and images used to form the training set is increased. The test set of 425 images is the same in all 4 trials

Both the SCG and CGB training algorithms exhibited "hit or miss" results with several runs yielding 1 or 2 100% misclassifications

Table 4.6: Summary of Trial 6

Trial Number = 7

Number of Ships = 3

Number of Classes = 3

Training Set = 360 images

Test Set = 425 images

Training/Test Image Folders = Exclusive

Trial Description = 3 Classes, 3 Ships, 2 Train Folders per Ship

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S200	20	60	0
H559S920	20	60	0
H559S480	20	0	123
H475S140	1	60	0
H068S032	1	60	0
H270S400	1	0	206
H056S030	2	60	0
H471S390	2	60	0
H472S420	2	0	96

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 20	Ship 1	Ship 2	
RP	100	3	97	0	54.5%	79.1%	76.0%	71.3%
SCG	100	3	97	0	76.4%	55.3%	86.5%	68.5%
GDX	100	0	0	100	82.9%	41.7%	88.5%	64.2%
CGB	100	14	86	0	80.5%	65.5%	99.0%	77.4%

Notes: Trials 6, 7, 8, and 9 measure the improvement in classification accuracies as the number of folders and images used to form the training set is increased. The test set of 425 images is the same in all 4 trials
Both the SCG and CGB training algorithms exhibited "hit or miss" results with several runs yielding 1 or 2 100% misclassifications

Table 4.7: Summary of Trial 7

Trial Number = 8

Number of Ships = 3

Number of Classes = 3

Training Set = 540 images

Test Set = 425 images

Training/Test Image Folders = Exclusive

Trial Description = 3 Classes, 3 Ships, 3 Train Folders per Ship

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S200	20	60	0
H559S920	20	60	0
H559S740	20	60	0
H559S480	20	0	123
H475S140	1	60	0
H068S032	1	60	0
H271S900	1	60	0
H270S400	1	0	206
H056S030	2	60	0
H471S390	2	60	0
H468S350	2	60	0
H472S420	2	0	96

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 20	Ship 1	Ship 2	
RP	100	1	99	0	76.4%	87.9%	94.8%	86.1%
SCG	100	0	99	1	83.7%	94.7%	95.8%	91.8%
GDX	100	0	0	100	83.7%	94.7%	95.8%	91.8%
CGB	100	1	99	0	84.6%	96.1%	93.8%	92.2%

Notes:

Trials 6, 7, 8, and 9 measure the improvement in classification accuracies as the number of folders and images used to form the training set is increased. The test set of 425 images is the same in all 4 trials

Both the SCG and CGB training algorithms exhibited "hit or miss" results with several runs yielding 1 or 2 100% misclassifications

Table 4.8: Summary of Trial 8

Trial Number = 9

Number of Ships = 3

Number of Classes = 3

Training Set = 720 images

Test Set = 425 images

Training/Test Image Folders = Exclusive

Trial Description = 3 Classes, 3 Ships, 4 Train Folders per Ship

Mean Squared Error Goal = 10^{-7}

Minimum Gradient Limit = 10^{-8}

Maximum Number of Epochs = 4000

Image Folder	Ship Number	Training Images	Test Images
H559S200	20	60	0
H559S920	20	60	0
H559S740	20	60	0
H559S780	20	60	0
H559S480	20	0	123
H475S140	1	60	0
H068S032	1	60	0
H271S900	1	60	0
H068S050	1	60	0
H270S400	1	0	206
H056S030	2	60	0
H471S390	2	60	0
H468S350	2	60	0
H056S090	2	60	0
H472S420	2	0	96

Training Algorithm	Number of Runs	Error Goal Reached	Minimum Derivative Reached	Maximum Epochs Reached	Best Result			Average Classification Accuracy
					Ship 20	Ship 1	Ship 2	
RP	100	0	100	0	81.3%	97.6%	78.1%	88.5%
SCG	100	0	100	0	90.2%	95.6%	91.7%	93.2%
GDX	100	0	0	100	79.7%	87.9%	90.6%	86.1%
CGB	100	0	100	0	88.6%	95.6%	92.7%	92.9%

Notes: Trials 6, 7, 8, and 9 measure the improvement in classification accuracies as the number of folders and images used to form the training set is increased. The test set of 425 images is the same in all 4 trials
Both the SCG and CGB training algorithms exhibited "hit or miss" results with several runs yielding 1 or 2 100% misclassifications

Table 4.9: Summary of Trial 9

set. In that respect, the latter trials were expected to represent the more realistic scenario for future applications of such a classifier in that attempts to classify an “unknown” target would almost certainly involve test images not previously seen (i.e., used in training) by the network classifier.

Trial 1, 2, and 3:

These three trials could be described as “proof of principle” trials in that they were conducted to prove that the basic premise that the Fourier spatial frequency feature vectors could, in fact, be used to distinguish between different ships was true. All three trials used a single folder per ship for both the training and test images. It is clear that the underlying pedagogy of using a Fourier spatial frequency to characterize the imagery is capable of distinguishing between ships with perfect or near perfect accuracy even in the case of the 13 ships used in trial 3.

The summary sheets show the classification accuracies for the “best result” from the 100 runs conducted. However, as such, it is important to note that even though a given training algorithm on a given trial may yield the same “best result” as an alternate algorithm, the behaviour over the entire set of 100 runs may be radically different. The notes on each summary page are intended to offer some indication of these differences. For example, a commonly observed behaviour for the algorithms SCG, GDX, and CGB (but never for RP) can be described as a “hit and miss” characteristic in which many runs result in one or more classes being 100% misclassified while the other classes are 100% (or nearly so) correctly classified. This characteristic is especially true of the GDX and CGB algorithms. Table 4.10 shows the classification rates for all 13 ships of trial 3 for each of the 100 runs for the training algorithm RP (the “best result”, run 60, is highlighted). This should be contrasted with the equivalent data in Table 4.11 for the training algorithm CGB (the “best result”, run 16, is highlighted). The latter shows several runs in which the network converged but resulted in as many as 10 classes being 100% misclassified (as zero). Nevertheless, the “best results” for both algorithms are essentially identical. It should also be noted that, in certain cases, the “best result” for a given algorithm and a trial of 100 runs is not necessarily a run which ultimately meets the minimum error criterion

Run No.	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12	Class 13	Average
1	86.4	100.0	100.0	100.0	93.9	95.5	94.0	100.0	100.0	94.3	92.0	94.1	100.0	96.2
2	87.7	98.1	100.0	98.1	87.8	100.0	96.0	95.0	100.0	96.2	96.0	97.1	100.0	96.5
3	90.1	96.2	100.0	100.0	90.9	100.0	98.0	100.0	100.0	96.2	100.0	100.0	100.0	97.8
4	91.4	98.1	100.0	100.0	93.9	100.0	98.0	100.0	100.0	98.1	94.0	100.0	100.0	98.0
5	92.6	100.0	100.0	100.0	93.9	100.0	96.0	100.0	97.0	96.2	98.0	94.1	100.0	97.5
6	88.9	100.0	100.0	100.0	93.9	95.5	96.0	100.0	100.0	94.3	94.0	100.0	96.0	96.8
7	91.4	100.0	100.0	98.1	90.9	95.5	98.0	100.0	97.0	96.2	100.0	97.1	96.0	96.9
8	90.1	100.0	92.3	98.1	90.9	100.0	94.0	100.0	97.0	96.2	100.0	100.0	100.0	96.8
9	92.6	100.0	100.0	100.0	100.0	100.0	98.0	95.0	100.0	98.1	100.0	91.2	96.0	97.8
10	91.4	100.0	100.0	100.0	97.0	100.0	92.0	100.0	100.0	98.1	100.0	97.1	96.0	97.8
11	88.9	98.1	100.0	100.0	93.9	100.0	94.0	100.0	100.0	98.1	96.0	94.1	100.0	97.2
12	86.4	100.0	100.0	100.0	93.9	100.0	100.0	95.0	100.0	100.0	98.0	94.1	96.0	97.2
13	85.2	100.0	96.2	100.0	97.0	100.0	96.0	100.0	97.0	98.1	96.0	97.1	96.0	96.8
14	87.7	100.0	96.2	98.1	87.9	100.0	98.0	100.0	100.0	98.1	98.0	100.0	96.0	96.9
15	91.4	100.0	100.0	100.0	97.0	95.5	98.0	100.0	100.0	88.7	90.0	97.1	100.0	96.7
16	88.9	100.0	96.2	100.0	90.9	100.0	100.0	95.0	100.0	88.7	96.0	100.0	96.0	96.3
17	91.4	98.1	100.0	100.0	93.9	95.5	100.0	100.0	100.0	98.1	96.0	97.1	96.0	97.5
18	90.1	98.1	100.0	98.1	97.0	95.5	96.0	95.0	100.0	92.5	100.0	100.0	100.0	97.1
19	88.9	98.1	100.0	100.0	93.9	100.0	96.0	100.0	100.0	96.2	100.0	97.1	100.0	97.7
20	88.9	98.1	100.0	98.1	97.0	100.0	98.0	95.0	100.0	96.2	98.0	100.0	100.0	97.8
21	93.8	100.0	100.0	100.0	97.0	100.0	98.0	95.0	100.0	98.1	96.0	100.0	96.0	98.0
22	91.4	98.1	96.2	98.1	87.9	100.0	90.0	100.0	100.0	98.1	100.0	94.1	96.0	96.4
23	90.1	94.3	100.0	100.0	97.0	100.0	98.0	100.0	100.0	98.1	92.0	94.1	100.0	97.2
24	88.9	100.0	96.2	96.2	87.9	100.0	96.0	95.0	100.0	92.5	96.0	100.0	100.0	96.0
25	95.1	98.1	96.2	98.1	97.0	100.0	96.0	100.0	100.0	96.2	98.0	100.0	100.0	98.0
26	90.1	100.0	100.0	100.0	93.9	100.0	96.0	100.0	100.0	98.1	96.0	100.0	100.0	98.2
27	90.1	98.1	80.8	100.0	90.9	95.5	94.0	100.0	100.0	96.2	100.0	97.1	96.0	95.3
28	92.6	100.0	96.2	100.0	97.0	100.0	98.0	100.0	100.0	90.6	100.0	97.1	96.0	97.5
29	88.9	98.1	100.0	100.0	93.9	100.0	98.0	100.0	100.0	96.2	96.0	100.0	96.0	97.5
30	96.3	100.0	96.2	100.0	97.0	100.0	100.0	100.0	100.0	100.0	86.0	97.1	100.0	97.9
31	90.1	100.0	100.0	100.0	90.9	100.0	100.0	100.0	100.0	96.2	96.0	97.1	100.0	97.7
32	88.9	100.0	100.0	98.1	93.9	100.0	98.0	100.0	97.0	96.2	98.0	100.0	100.0	97.7
33	90.1	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	98.1	96.0	94.1	96.0	97.9
34	90.1	98.1	96.2	100.0	90.9	95.5	100.0	100.0	100.0	100.0	98.0	100.0	100.0	97.6
35	92.6	98.1	96.2	100.0	90.9	95.5	96.0	100.0	97.0	96.2	94.0	94.1	100.0	96.2
36	90.1	98.1	92.3	98.1	100.0	100.0	98.0	100.0	97.0	94.3	94.0	97.1	100.0	96.8
37	91.4	100.0	100.0	100.0	97.0	95.5	98.0	100.0	100.0	100.0	100.0	97.1	100.0	98.4
38	93.8	98.1	100.0	100.0	97.0	100.0	96.0	100.0	100.0	94.3	96.0	100.0	100.0	98.1
39	91.4	100.0	100.0	100.0	93.9	100.0	98.0	100.0	100.0	98.1	96.0	88.2	96.0	96.9
40	92.6	98.1	100.0	100.0	90.9	100.0	98.0	100.0	100.0	98.1	96.0	97.1	100.0	97.9
41	90.1	96.2	100.0	93.9	100.0	93.9	100.0	94.0	100.0	97.0	92.5	94.1	96.0	96.2
42	86.4	100.0	100.0	98.1	87.9	100.0	96.0	100.0	97.0	96.2	96.0	94.1	100.0	96.3
43	91.4	100.0	100.0	98.1	93.9	100.0	98.0	100.0	100.0	83.0	90.0	100.0	96.0	96.2
44	90.1	98.1	100.0	100.0	90.9	100.0	98.0	100.0	100.0	96.2	96.0	97.1	96.0	97.1
45	93.8	100.0	92.3	98.1	97.0	100.0	98.0	100.0	100.0	100.0	96.0	94.1	96.0	97.3
46	85.2	98.1	96.2	100.0	87.9	100.0	94.0	95.0	100.0	92.5	100.0	100.0	100.0	96.1
47	90.1	98.1	100.0	100.0	97.0	100.0	98.0	100.0	100.0	96.2	96.0	100.0	96.0	97.8
48	93.8	100.0	100.0	100.0	100.0	95.5	98.0	100.0	100.0	96.2	100.0	100.0	96.0	98.4
49	93.8	96.2	100.0	100.0	93.9	100.0	100.0	95.0	100.0	96.2	94.0	100.0	96.0	97.3
50	87.7	100.0	100.0	94.2	93.9	95.5	96.0	95.0	97.0	98.1	96.0	97.1	100.0	96.2
51	91.4	98.1	100.0	98.1	90.9	100.0	98.0	100.0	100.0	86.8	96.0	94.1	100.0	96.4
52	91.4	96.2	100.0	100.0	90.9	100.0	96.0	100.0	97.0	98.1	96.0	91.2	100.0	96.7
53	93.8	98.1	100.0	100.0	90.9	100.0	98.0	100.0	97.0	96.2	100.0	100.0	96.0	97.7
54	95.1	100.0	100.0	100.0	100.0	100.0	100.0	95.0	100.0	98.1	96.0	94.1	96.0	98.2
55	93.8	100.0	100.0	100.0	90.9	100.0	98.0	100.0	97.0	92.5	100.0	100.0	100.0	97.9
56	93.8	98.1	100.0	100.0	93.9	100.0	96.0	95.0	97.0	96.2	98.0	94.1	100.0	96.3
57	88.9	100.0	96.2	100.0	97.0	100.0	96.0	100.0	100.0	96.2	100.0	100.0	100.0	98.0
58	93.8	96.2	100.0	100.0	90.9	100.0	98.0	100.0	97.0	96.2	96.0	100.0	96.0	97.4
59	91.4	92.9	96.3	100.0	93.9	100.0	94.1	95.0	100.0	96.2	100.0	100.0	100.0	96.9
60	93.8	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	96.4
61	91.4	100.0	100.0	100.0	100.0	95.5	98.0	100.0	100.0	98.1	92.0	100.0	100.0	98.1
62	92.6	96.2	100.0	98.1	93.9	100.0	98.0	100.0	100.0	98.1	96.0	100.0	100.0	97.9
63	91.4	98.1	100.0	100.0	87.9	100.0	92.0	100.0	100.0	94.3	100.0	100.0	96.0	96.9
64	87.7	100.0	100.0	98.1	97.0	100.0	100.0	100.0	100.0	96.2	96.0	94.1	96.0	97.3
65	92.6	100.0	96.2	100.0	97.0	100.0	100.0	100.0	100.0	96.2	96.0	97.1	96.0	97.9
66	87.7	98.1	96.2	100.0	93.9	100.0	98.0	100.0	100.0	96.2	90.0	100.0	100.0	96.9
67	90.1	100.0	96.2	100.0	93.9	100.0	98.0	100.0	100.0	100.0	94.0	100.0	100.0	97.9
68	91.4	98.1	100.0	100.0	90.9	100.0	96.0	100.0	97.0	96.2	96.0	97.1	100.0	97.1
69	86.4	100.0	96.2	100.0	100.0	100.0	96.0	100.0	100.0	96.2	96.0	97.1	100.0	97.7
70	92.6	98.1	100.0	100.0	90.9	100.0	96.0	100.0	100.0	100.0	100.0	97.1	100.0	98.1
71	86.4	100.0	96.2	98.1	93.9	100.0	96.0	100.0	100.0	88.8	96.0	94.1	84.0	94.7
72	91.4	100.0	100.0	100.0	87.9	95.5	100.0	95.0	100.0	100.0	100.0	97.1	100.0	97.4
73	92.6	100.0	96.2	100.0	97.0	100.0	98.0	100.0	100.0	100.0	98.0	97.1	100.0	98.4
74	91.4	100.0	100.0	100.0	93.9	100.0	96.0	100.0	100.0	92.5	96.0	97.1	100.0	97.4
75	87.7	92.5	100.0	100.0	93.9	100.0	98.0	100.0	100.0	96.2	96.0	97.1	100.0	97.2
76	88.9	98.1	100.0	98.1	97.0	100.0	100.0	95.0	100.0	96.2	96.0	97.1	96.0	97.3
77	90.1	100.0	100.0	100.0	93.9	100.0	100.0	100.0	100.0	94.3	98.0	97.1	100.0	98.0
78	93.8	98.1	100.0	100.0	100.0	100.0	98.0	100.0	100.0	98.1	98.0	94.1	100.0	98.5
79	88.9	98.1	100.0	98.1	93.9	95.5	94.0	100.0	97.0	94.3	96.0	100.0	100.0	96.6
80	92.6	98.1	100.0	98.1	87.9	95.5	96.0	95.0	100.0	98.1	96.0	97.1	100.0	96.6
81	95.1	100.0	100.0	100.0	90.9	95.5	96.0	100.0	100.0	100.0	98.0	100.0	100.0	98.1
82	92.6	98.1	96.2	100.0	87.9	95.5	94.0	100.0	100.0	94.3	96.0	100.0	96.0	95.4
83	90.1	100.0	96.2	96.2	93.9	100.0	98.0	100.0	100.0	98.1	100.0	97.1	96.0	97.3
84	87.7	98.1	92.3	100.0	97.0	100.0	98.0	95.0	100.0	100.0	94.0	100.0	100.0	97.1
85	93.8	98.1	100.0	100.0	97.0	100.0	96.0	100.0	100.0	94.3	98.0	100.0	100.0	98.3</

Run No.	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12	Class 13	Average
1	93.8	0.0	100.0	100.0	0.0	0.0	100.0	0.0	100.0	98.1	100.0	0.0	100.0	80.9
2	0.0	100.0	98.2	0.0	100.0	100.0	0.0	0.0	97.0	100.0	100.0	97.1	0.0	80.8
3	0.0	100.0	100.0	0.0	0.0	100.0	98.0	100.0	100.0	0.0	0.0	100.0	100.0	81.4
4	92.8	100.0	100.0	0.0	0.0	100.0	98.0	0.0	100.0	100.0	100.0	0.0	100.0	88.5
5	0.0	100.0	100.0	100.0	97.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	91.9
6	95.1	100.0	100.0	100.0	93.9	100.0	100.0	100.0	97.0	100.0	100.0	100.0	100.0	98.9
7	92.8	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	98.1
8	95.1	100.0	100.0	100.0	97.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	99.2
9	92.8	100.0	100.0	100.0	90.9	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.3
10	95.1	0.0	100.0	100.0	100.0	100.0	98.0	100.0	97.0	98.1	100.0	100.0	100.0	91.4
11	93.8	100.0	100.0	100.0	100.0	100.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	91.7
12	92.8	100.0	100.0	100.0	97.0	100.0	96.0	100.0	100.0	0.0	98.0	100.0	100.0	91.0
13	92.8	100.0	100.0	100.0	97.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	99.0
14	97.5	100.0	0.0	100.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	100.0	0.0	38.3
15	90.1	98.1	100.0	100.0	97.0	100.0	98.0	100.0	100.0	100.0	98.0	97.1	100.0	98.3
16	95.1	100.0	100.0	100.0	97.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.4
17	97.5	0.0	100.0	100.0	0.0	0.0	100.0	100.0	100.0	0.0	100.0	100.0	0.0	61.3
18	0.0	5.7	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	0.0	100.0	77.2
19	95.1	100.0	100.0	0.0	97.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	76.3
20	92.8	100.0	100.0	100.0	97.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	91.5
21	93.8	98.1	100.0	0.0	100.0	100.0	98.0	100.0	0.0	0.0	0.0	97.1	100.0	88.1
22	92.8	100.0	100.0	100.0	97.0	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.8
23	92.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	100.0	98.1	0.0	100.0	100.0	91.0
24	93.8	100.0	100.0	100.0	93.9	100.0	100.0	100.0	100.0	98.1	100.0	100.0	100.0	98.9
25	92.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	97.0	98.1	100.0	100.0	100.0	98.3
26	91.4	0.0	100.0	15.4	97.0	0.0	0.0	0.0	0.0	0.0	97.1	100.0	0.0	46.2
27	95.1	100.0	100.0	100.0	97.0	0.0	100.0	0.0	100.0	100.0	100.0	0.0	100.0	78.5
28	95.1	100.0	0.0	100.0	97.0	100.0	98.0	100.0	98.1	100.0	100.0	100.0	100.0	91.4
29	92.8	100.0	100.0	0.0	97.0	100.0	0.0	100.0	100.0	100.0	100.0	0.0	100.0	78.1
30	93.8	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	98.1	100.0	100.0	100.0	88.6
31	93.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.7
32	92.8	100.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	100.0	37.9
33	93.8	100.0	100.0	100.0	100.0	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	99.1
34	93.8	100.0	100.0	100.0	100.0	100.0	92.0	100.0	100.0	100.0	100.0	100.0	100.0	98.9
35	95.1	100.0	100.0	100.0	100.0	0.0	100.0	100.0	0.0	0.0	100.0	0.0	100.0	88.9
36	91.4	100.0	0.0	100.0	97.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	100.0	60.6
37	91.4	100.0	100.0	100.0	0.0	100.0	98.0	0.0	100.0	100.0	100.0	100.0	100.0	83.8
38	92.8	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	98.1	100.0	100.0	100.0	98.1
39	92.8	100.0	100.0	100.0	93.9	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	75.9
40	0.0	100.0	100.0	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	0.0	53.8
41	95.1	100.0	0.0	100.0	97.0	100.0	0.0	0.0	0.0	97.0	0.0	100.0	100.0	60.7
42	92.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	98.7
43	88.9	100.0	100.0	100.0	90.9	100.0	98.0	95.0	100.0	100.0	100.0	100.0	100.0	97.9
44	91.4	100.0	100.0	100.0	93.9	0.0	98.0	100.0	100.0	96.2	100.0	100.0	100.0	90.7
45	90.1	100.0	100.0	100.0	97.0	100.0	100.0	0.0	97.0	100.0	0.0	0.0	100.0	75.7
46	97.5	100.0	0.0	100.0	97.0	0.0	98.0	100.0	100.0	98.1	100.0	100.0	100.0	93.9
47	92.8	100.0	100.0	100.0	93.9	95.5	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.1
48	93.8	100.0	100.0	0.0	97.0	95.5	96.0	100.0	0.0	92.5	100.0	0.0	100.0	75.0
49	97.5	100.0	100.0	100.0	97.0	0.0	0.0	100.0	100.0	98.1	98.0	100.0	100.0	83.9
50	90.1	100.0	100.0	100.0	97.0	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.5
51	96.3	100.0	0.0	100.0	0.0	0.0	100.0	0.0	100.0	0.0	100.0	100.0	100.0	81.3
52	0.0	0.0	100.0	0.0	0.0	0.0	100.0	100.0	100.0	98.1	100.0	100.0	100.0	81.4
53	92.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	100.0	98.1	100.0	100.0	100.0	98.7
54	92.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	98.8
55	91.4	0.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	0.0	98.0	97.1	100.0	85.4
56	91.4	100.0	100.0	100.0	100.0	100.0	98.0	0.0	100.0	100.0	100.0	100.0	100.0	91.5
57	97.5	100.0	100.0	100.0	100.0	0.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	92.0
58	95.1	100.0	0.0	0.0	90.9	100.0	0.0	100.0	97.0	100.0	100.0	0.0	100.0	67.9
59	91.4	98.1	100.0	100.0	90.9	95.5	98.0	100.0	97.0	100.0	100.0	94.1	100.0	97.3
60	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	82.4	0.0	14.2
61	91.4	100.0	0.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	91.3
62	93.8	100.0	100.0	100.0	93.9	0.0	98.0	100.0	100.0	98.1	100.0	100.0	100.0	91.1
63	91.4	100.0	100.0	100.0	93.9	100.0	98.0	0.0	97.0	98.1	100.0	100.0	100.0	90.8
64	91.4	100.0	100.0	100.0	100.0	100.0	98.0	95.0	100.0	100.0	100.0	100.0	100.0	98.8
65	92.8	100.0	100.0	100.0	97.0	95.5	0.0	100.0	97.0	100.0	100.0	100.0	100.0	90.9
66	92.8	100.0	100.0	100.0	93.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.0	81.3
67	92.8	100.0	100.0	100.0	97.0	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.8
68	0.0	100.0	0.0	0.0	93.9	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	53.4
69	92.8	0.0	100.0	100.0	100.0	100.0	98.0	100.0	97.0	96.2	100.0	97.1	100.0	90.8
70	0.0	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	0.0	84.5
71	91.4	100.0	100.0	100.0	97.0	100.0	98.0	100.0	0.0	100.0	100.0	100.0	100.0	91.1
72	12.9	0.0	0.0	0.0	13.2	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	9.7
73	92.8	100.0	100.0	100.0	97.0	100.0	0.0	100.0	97.0	100.0	100.0	100.0	100.0	91.3
74	0.0	100.0	100.0	100.0	97.0	100.0	0.0	100.0	0.0	100.0	100.0	100.0	100.0	76.7
75	91.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	98.3
76	92.8	100.0	100.0	100.0	100.0	100.0	98.0	100.0	97.0	100.0	100.0	97.1	100.0	98.8
77	91.4	0.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	91.5
78	90.1	100.0	100.0	100.0	90.9	100.0	98.0	100.0	100.0	100.0	100.0	100.0	100.0	98.4
79	92.8	100.0	100.0	100.0	93.9	100.0	98.0	100.0	97.0	100.0	100.0	100.0	100.0	98.6
80	92.8	0.0	0.0	100.0	100.0	100.0	98.0	95.0	97.0	100.0	0.0	97.1	100.0	75.4
81	0.0	100.0	100.0	100.0	100.0	100.0	98.0	100.0	100.0	0.0	100.0	100.0	100.0	84.5
82	93.8	100.0	100.0	100.0	97.0	100.0	100.0	100.0	97.0	100.0	100.0	97.1	100.0	98.8
83	95.1	0.0	100.0	100.0	100.0	100.0	98.0	100.0	97.0	100.0	0.0	100.0	100.0	83.8
84	92.8	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	100.0	0.0	100.0	68.7
85	91.4	98.1	0.0	0.0	0.0	100.0	100.0	0.0	97.0	98.1	0.0	0.0	0.0	45.0
86	0.0													

described in the previous section. In certain cases, the “best result” run was one terminated on either reaching the minimum error derivative or the maximum number of training epochs. Indeed, many of the runs listed in Table 4.11 which resulted in completely misclassified classes nevertheless trained and met the minimum network error criterion. Table 4.11 (and, to a lesser extent, Table 4.10) also illustrates the essential need to carry out multiple training runs for any given training/test environment since the results from run-to-run can vary dramatically depending upon the training algorithm used. For example, contrast run 16 in Table 4.11 with the result of run 72 in which essentially only one of the 13 classes is correctly identified. In practice, even though both the RP and CGB algorithms yielded the same “best result” in this case, the consistency of the RP data in Table 4.10 would favour its adoption over the CGB algorithm since the data would strongly indicate that the RP algorithm is more strongly “matched” to this particular classification task than the CGB method.

Trials 4 and 5:

These two trials again used “common” train/test folders but with three folders per ship. The training and test sets were identical in each of these trials, the difference lying in the choice of definition of the classes. In trial 4, each folder is considered a class (nine classes in all, one per folder) while in trial 5, the three folders for each ship are collectively considered a class (three classes in all, one per ship). Together, these two trials provide a measure or comparison of the ability for the network classifier to distinguish between same-ship folders for multiple ships as opposed to the ability to distinguish between ships represented by multiple folders per ship. The results for trial 5 are noticeably better than those of trial 4 and thus confirm, (a), that there is a high degree of correlation between disparate folders for a given ship (poorer discrimination between classes in trial 4) and, (b), the ability to accurately recognize and correctly classify a given ship improves as the number of disparate folders and total number of images used in the training process is increased.

Trials 6 to 10:

These trials differ significantly from the previous trials in that the training and test sets are “exclusive”, i.e., the test set images are drawn from folders not used in the composition of the training sets. The same test set is used in each trial while additional folders for the individual ships are added to the training set in each succeeding trial, starting with one training folder in trial 6 and increasing to four folders in trial 9. As expected and as shown in Figure 4.1, the overall classification accuracy of the network classifier improves for each of the training algorithms as the number of training folders is increased. It would be desirable to pursue this scheme to include a much larger number of training folders (an objective which is stymied by the lack of data in the image database) since it must not be assumed that this improvement in overall classification accuracy would monotonically increase. As more and more folders representing a wider variety of views of the ships are added to the training set, a measure of degeneracy is expected to arise owing to the fact that, for certain views of the ships (e.g., low attitude and/or near-stern or near-bow views), the images for different ships would be essentially indistinguishable. Thus the addition of such folders to the training set would be expected to limit or even reduce overall classification accuracies. It is possible, therefore, in practice, that the training set size for a working, trained neural network classifier would have to be limited and, more importantly, contain a balanced mix of folders representing non-redundant but comprehensive ISAR views of the ship.

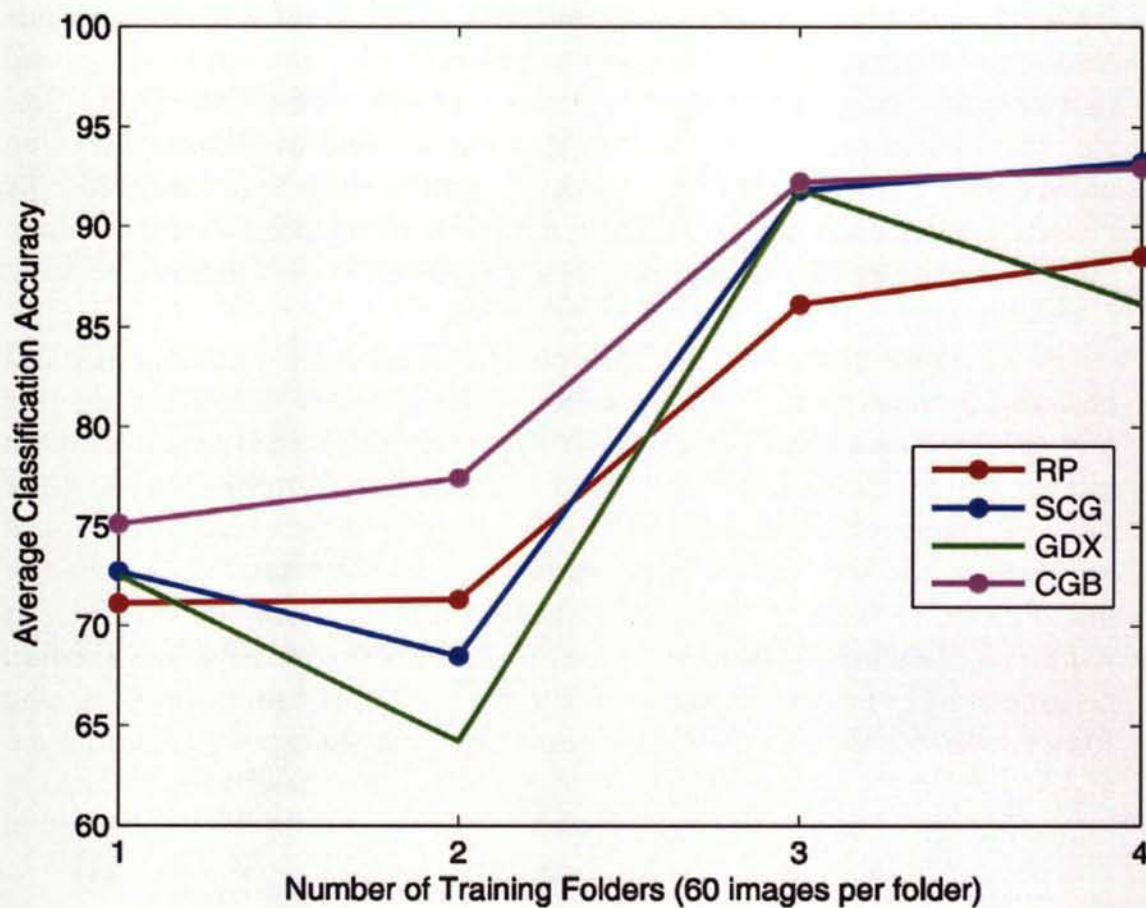


Figure 4.1: Improvement in classification accuracies with number of training folders

5: Concluding Remarks

The experimental results offered in the trials described in chapter 4 confirm the viability and soundness of the premise that characterizing these types of ISAR images by a Fourier spatial frequency spectrum leads to accurate, reproducible image recognition using an artificial neural network classifier. The underlying idea that the 'human classifier rules' can be encompassed in a Fourier spatial spectrum and that an appropriately designed and trained neural network can then 'discover' the essential relationships in the spectral components needed to yield accurate pattern recognition is a valid one.

The results from trials 1 through 5 illustrated convincingly that perfect or near-perfect classification rates can be obtained by the methodology described in this report when the classifier is presented with imagery from the same folders used in the training process even in the case of several ships as in trial 3. The results from trials 6 through 9 represented the more realistic scenario of a classifier presented with test images drawn from image folders not seen in the training phase. As expected, the classification rates were generally lower but clearly could be improved to fall in the 80% or 90% range when the number of training folders used for the different ships was increased.

There are three principal areas of work which should be pursued in order to further the research effort presented here.

- (1) There is an obvious need for additional, more methodical data. Although the initial image database utilized in the present study was impressively large, it was clear that the distribution of imagery was highly askew and that, after vetting of the images, many of the ships contained in the database were poorly represented. As the work in this report has revealed, it is necessary to have not only many images of a given ship but to have these images distributed over many different folders, the latter particular to one 'view' of the ISAR platform. For future studies of automated classification, it is highly desirable to generate an image database for a given ship in a more

methodical manner by systematically collecting images corresponding to a prescribed regimen of angles of view of the imaging process. In this way, the data used for training the classifier would present a much more comprehensive and complete representation of the ship's image describing the ISAR view from different (and hopefully, known) angles of approach. To this end, such a dataset would prove valuable even if only a small number of ships were so studied, i.e., a all-inclusive set of ship image folders for a small number of ships would better enable the design and testing of a classification methodology than a much larger dataset of a larger number of ships for which only limited or redundant views are available.

- (2) There is a need to enhance the feature vector representation of an image to include, in addition to the Fourier spatial frequency components, one or more physical attributes of the ship determined from the image. The example of estimating the overall ship length has been discussed in section 3.3, a discussion which also served to illustrate the innate difficulties that may be encountered in such a task. Nevertheless, incorporating into the image feature vector a quantity such as ship length may prove an essential requisite to the design and implementation of a classification system intended to operate for a large number of disparate ships and ship types.
- (3) An important enhancement to the fundamental approach of feature vector derivation and neural network classification would be the exploitation of the 'ciné' nature of the ISAR imagery (Musman S., Kerr D., and Bachmann C., 1996). The present work has treated the imagery of a ship one image at a time without any reference to the other, generally similar, images contained in the same data folder. The intrinsic nature of the collection of ISAR images leads to datasets consisting of image folders (one flight and scene number) containing up to several hundred individual images. Thus, as illustrated in section 3.3, while any given image may produce an incorrect estimate of overall ship length, the measurement of such a parameter over the *sequence* of images can lead to a consistent

and more accurate estimate by exploiting the similarity of the individual images and by objectively assessing the quality of these images. However, it cannot be assumed that exploitation of the plurality of images within an image folder is merely a question of averaging or accumulation of estimates or of classification results. Statistical inference would certainly play a role here along with some methodology of determining which images are superior to others in the sequence, i.e., a weighting of the results. Such an approach would also most likely involve reiteration of the image sequence in that images would be revisited and analyzed once the features have been calculated and evaluated over the full sequence. The “final” version of the feature vector for any given individual image would then be influenced by the measurements of the feature vectors for all of the images in the test image folder. Finally, the conclusion of the classifier as to the identity of the ship for a given folder would be formed by a “committee approach” in that the classification results for each image would be weighted and statistically combined to yield a single, concluding classification result.

References

English, R.A., "Automatic Target Recognition Using HNeT", Technical Memorandum DREO TM 2001-80, Defence R&D Canada, December 2001.

Gagnon L. and Klepko R., "Hierarchical Classifier Design for Airborne SAR Images of Ships", SPIE Proc. 3371 of conference "Automatic Target Recognition VIII", Orlando, 1998.

Gouaillier V. and Gagnon L., "Ship Silhouette Recognition Using Principal Components Analysis", SPIE Proc. 3164 of conference "Applications of Digital Image Processing XX", San Diego, 1997.

Haykin, Simon, "Neural Networks: A Comprehensive Foundation", 2nd edition, Prentice Hall, 1998.

MatLab Neural Network Toolbox Manual, version 4, MathWorks, 2000.

Musman S., Kerr D., and Bachmann C., "Automatic Recognition of ISAR Ship Images", IEEE Trans. Aerospace and Electronic Systems, Vol. 32, No. 4, pp. 1392-1403, Oct. 1996.

Tessier Y. and Shahbazian E., "Evaluation of Decision Rule Based Neural Net on SpotSAR Imagery of Military Ships", Technical Report Contract No. W7714-9-0254/38SV, prepared by Lockheed Martin Canada for Defense Research Establishment Ottawa, September 2000.

Tessier Y. and Shahbazian E., Annex A of "Evaluation of Decision Rule Based Neural Net on SpotSAR Imagery of Military Ships", Technical Report Contract No. W7714-9-0254/38SV, prepared by Lockheed Martin Canada for Defense Research Establishment Ottawa, September 2000.

Appendix A

Listing of the MatLab algorithm 'fingerprint.m'

```
% FINGERPRINT.M
%
% Author: K. L. Sala, Communications Research Center, Ottawa
% Last Revised:: September 2005
%
% This routine performs the following series of operations upon an image from the ProSpot database of ship ISAR
% images:
%
% 1. List the ship folders within a particular directory to allow the user to select (by number) a specific folder;
% 2. List the image files within the folder chosen in step 1 to allow the user to select one specific ship image;
% 3. Read the image file and header information and normalize the image intensity values to max = 1 (all of the files
%    initially have initial max = 65535);
% 4. Apply a high-intensity-pass filter to the image to increase the contrast and reduce the amount of background
%    noise;
% 5. Perform an edge detection operation (Sobel) on the filtered image;
% 6. Perform a Radon transform on the edge detected image;
% 7. Crop the Radon transform to exclude any detected lines close to  $\pm 90^\circ$ ;
% 8. Determine the maximum value of the Radon transformation, rotate the image so that the principal axis of the
%    ship
%    image is horizontal and determine the new image dimensions of the rotated image;
% 9. Measure the vertical profile of the image (i.e., a row vector of the image integrated along the horizontal),
%    determine
%    the peak of the profile, and subsequently crop the rotated image to eliminate any vertical ghosting and/or
%    spurious
%    images in the image;
% 10. Determine the (horizontal) profile of the filtered, rotated, and cropped image;
% 11. Approximately center the ship profile and pad it to a dimension of 512 (for FFT purposes);
% 12. Calculate the FFT of the centered, padded ship profile;
% 13. Zero the DC component of the FFT;
% 14. Cut the FFT spectrum to a pre-determined length of "specdim" specified at the start of this routine;
% 15. Normalize the cut FFT spectrum to 1;
% 16. Display various MatLab figures illustrating the various stages above.
%
%=====
===
%
% Initialize the MatLab workspace and define the parameters 'specdim' and 'root'.
%
clear all; close all; clc;
specdim=64;
root='d:\prospot_codenames\';
cd(root);
%
%=====
===
%
% Begin by listing the folders found at the path 'root' and prompt user to select one folder (by directory number).
%
LD=dir("*.");
[NED,NDUMD]=size(LD);
NED=NED-2;
lstD=[1:NED];
nr1stD=fix(NED/4)+1;
nrem1stD=NED-4*(fix(NED/4));
fprintf('Directories by number : \n')
if nr1stD > 1
    for i=1:(nr1stD-1),
        j=(i-1)*4;
        fprintf('%-3i  %-18s  %-3i  %-18s  %-3i  %-18s  %-3i  %-18s\n',lstD(j+1),LD(j+3).name,lstD(j+2),LD(j+4).name,...
```

```

        lstD(j+3),LD(j+5).name,lstD(j+4),LD(j+6).name)
    end
end
j=4*(nrlist-1);
if nremlst==3
    fprintf('%-3i %-18s %-3i %-18s %-3i %-18s\n',lstD(j+1),LD(j+3).name,lstD(j+2),LD(j+4).name,...
        lstD(j+3),LD(j+5).name)
elseif nremlst==2
    fprintf('%-3i %-18s %-3i %-18s\n',lstD(j+1),LD(j+3).name,lstD(j+2),LD(j+4).name)
elseif nremlst==1
    fprintf('%-3i %-18s\n',lstD(j+1),LD(j+3).name)
end
dirname = input('\nEnter directory number : ','s');
dimnum=str2num(dirname);
longdirname=LD(dimnum+2).name;
subdir=strcat(root,longdirname,'\');
cd(subdir);
%
% Now pointing to the chosen ship directory. List the image files within the folder and prompt user to pick a specific
file
% by number.
%
L=dir('*. *');
[NE,NDUM]=size(L);
NE=NE-2;
lst=[1:NE];
nrlist=fix(NE/4)+1;
nremlst=NE-4*(fix(NE/4));
fprintf('Directory entries by number : \n')
if nrlist > 1
    for i=1:(nrlist-1),
        j=(i-1)*4;
        fprintf('%-3i %-18s %-3i %-18s %-3i %-18s %-3i %-18s\n',lst(j+1),L(j+3).name,lst(j+2),L(j+4).name,...
            lst(j+3),L(j+5).name,lst(j+4),L(j+6).name)
    end
end
j=4*(nrlist-1);
if nremlst==3
    fprintf('%-3i %-18s %-3i %-18s %-3i %-18s\n',lst(j+1),L(j+3).name,lst(j+2),L(j+4).name,...
        lst(j+3),L(j+5).name)
elseif nremlst==2
    fprintf('%-3i %-18s %-3i %-18s\n',lst(j+1),L(j+3).name,lst(j+2),L(j+4).name)
elseif nremlst==1
    fprintf('%-3i %-18s\n',lst(j+1),L(j+3).name)
end
filenumber = input('\nEnter file number : ','s');
filenum=str2num(filenumber);
longfilename=L(filenum+2).name;
fullfilename=strcat('d:\prospot\ ',longdirname,'\ ',longfilename);
fid=fopen(longfilename,'r','b');
%
%=====
%
%
% Now have chosen an image file from within a chosen ship folder. Open the file and read the image header
information.
%
fileid=fread(fid,1,'ulong');imageheadersize=fread(fid,1,'ulong');
imagefilever=fread(fid,1,'ulong');imagedataformat=fread(fid,1,'ulong');
pixelspaceangle=fread(fid,1,'float');pixelspaceazimuth=fread(fid,1,'float');
pixelresrange=fread(fid,1,'float');pixelresazimuth=fread(fid,1,'float');
sarmode=fread(fid,1,'ulong');sartapevolno=fread(fid,1,'ulong');
missionid=fread(fid,8,'char');
done=8;
for k=1:8,
    if missionid(k)==0

```



```

done=k-1;
end
end
if done==0
done=1;
end
mission=sprintf('%i',missionid(1:done));firstsceneno=fread(fid,1,'ulong');
targetselectmode=fread(fid,1,'ulong');receivergain=fread(fid,1,'ulong');
acheading=fread(fid,1,'float');acaltitude=fread(fid,1,'ulong');
inversefilter=fread(fid,1,'ulong');lfmrate=fread(fid,1,'float');
squintangle=fread(fid,1,'float');velocity=fread(fid,1,'float');
prf=fread(fid,1,'float');samplingrate=fread(fid,1,'float');
adaption=fread(fid,1,'ulong');lfft=fread(fid,1,'ulong');
aperturetime=fread(fid,1,'float');overlap=fread(fid,1,'float');
nramp=fread(fid,1,'ulong');autofocus=fread(fid,1,'ulong');
secondaryname=fread(fid,44,'char');
done=44;
for k=1:44,
if secondaryname(k)==0
done=k-1;
end
end
if done==0
done=1;
end
secondname=sprintf('%s',char(secondaryname(1:done)));
numberpulses=fread(fid,1,'ulong');
rcmc=fread(fid,1,'ulong');rangecorrect=fread(fid,1,'ulong');
trackanglestart=fread(fid,1,'float');xyaltitude=fread(fid,1,'float');
%
% Now read the parameters contained in the frame header
%
frewind(fid);
status=fseek(fid,imageheadersize,-1);
frameheadersize=fread(fid,1,'ulong');maxpixelvalue=fread(fid,1,'float');
rangeofmax=fread(fid,1,'ulong');azimuthofmax=fread(fid,1,'ulong');
latitude=fread(fid,1,'ulong');longitude=fread(fid,1,'ulong');
rangeframe=fread(fid,1,'ulong');timestamp=fread(fid,1,'ulong');
scenenummer=fread(fid,1,'ulong');lastframe=fread(fid,1,'ulong');
numbercolumns=fread(fid,1,'ulong');numberrows=fread(fid,1,'ulong');
framescalefactor=fread(fid,1,'float');radialspeed=fread(fid,1,'float');
frewind(fid);
%
% Now read the number of rows and columns from the frame header.
%
status=fseek(fid,2088,-1);
N=fread(fid,2,'uint');
%
% Take care on this point! The first integer read, N(1), is the number of COLUMNS while N(2) is the number of
ROWS.
%
frewind(fid);
status=fseek(fid,2096,-1);
FSF=fread(fid,1,'float');
frewind(fid);
status=fseek(fid,4096,-1);
%
% Now read the actual image data in matrix form. Note that max(max(iraw)) = 65535.
% Normalize I such that max(max(I)) = 1.0
%
I=fread(fid,[N(1),N(2)],'ushort');iraw=I;
fclose(fid);
I=I/65535;A=I;
%
% Now apply a filter function to the image intensity in the form
%
```



```

% I = factor*I where factor = 1 - exp(-alpha*(I^2)) with
%
% alpha = -100*ln(1 - OTL) where OTL is the transmission factor for
% the 'one tenth level'
% i.e., OTL = 1 - exp(-0.01*alpha)
%
OTL=0.10;
alpha=100*log(1/(1-OTL));
for j=1:N(1),;
    for i=1:N(2),;
        factor=1.-exp(-alpha*(I(i,j))^2);
        F(i,j)=factor;
        I(i,j)=factor*I(i,j);
    end;
end;
%
% Display the header information.
%
clc;
fprintf('Processing Image %s Columns x %i Rows\n',fullfilename,N(1),N(2))
fprintf('Image Header Information\n');
fprintf('pixelspaceazimuth = %6.2f,pixelspaceazimuth = %6.2f,pixelspaceazimuth = %6.2f,pixelspaceazimuth = %6.2f,pixelspaceazimuth = %6.2f\n',
pixelspaceazimuth,pixelspaceazimuth,pixelspaceazimuth,pixelspaceazimuth,pixelspaceazimuth);
fprintf('imagefilever = %6.2f,imagefilever = %6.2f,imagefilever = %6.2f,imagefilever = %6.2f,imagefilever = %6.2f\n',
imagefilever,imagefilever,imagefilever,imagefilever,imagefilever);
fprintf('imageheadersize = %6i,rangeframe = %6i,imageheadersize,rangeframe);
fprintf('sarmode = %6.2f,sartapevolno = %6.2f,sarmode,sartapevolno);
fprintf('targetselectmode = %6.2f,numberpulses = %6i,targetselectmode,numberpulses);
fprintf('acheading,acaltitude = %6.2f,acheading,acaltitude);
fprintf('lfrate,samplingrate = %6.2f,lfrate,samplingrate);
fprintf('squintangle,velocity = %6.2f,squintangle,velocity);
fprintf('adaption,lfft = %6.2f,adaption,lfft);
fprintf('aperturetime,overlap = %6.2f,aperturetime,overlap);
fprintf('nramp,autofocus = %6.2f,nramp,autofocus);
fprintf('rangecorrect,trackanglestart = %6.2f,rangecorrect,trackanglestart);
fprintf('xyaltitude,radialspeed = %6.2f,xyaltitude,radialspeed);
fprintf('frameheadersize,firstsceneno = %6i,frameheadersize,firstsceneno);
fprintf('rangeofmax,azimuthofmax = %6.2f,rangeofmax,azimuthofmax);
fprintf('latitude,longitude = %11i,latitude,longitude);
fprintf('scenenum,frame = %6i,scenenum,frame);
fprintf('numbercolumns,rows = %6i,numbercolumns,rows);
fprintf('missionid = %3i %3i %3i %3i %3i %3i %3i %3i,missionid(1:5));
fprintf('inversefilter = %16i,inversefilter);
fprintf('prf = %16i,prf);
fprintf('receivergain = %16i,receivergain);
fprintf('rcmc = %16i,rcmc);
fprintf('timestamp = %16i,timestamp);
fprintf('maxpixelvalue = %16e,maxpixelvalue);
fprintf('framescalefactor = %16i,framescalefactor);
fprintf('Product : MaxPixelValue x FrameScaleFactor = %10.2f,(maxpixelvalue*framescalefactor));
fprintf('\n');
%
% Now carry out an edge detection of the raw image followed by a Radon transform. Then rotate the filtered image
% clockwise by -(thetamax-90) degrees.
%
BW=edge(I,'sobel');
theta=1:179;
[R,yp]=radon(BW,theta);
%
% To avoid the case where there is a strong vertical line within the image which will be the max for the Radon
matrix,
% we (ad hoc) 'cut' out the beginning and end portions of the Radon matrix R (i.e., we eliminate any rotation angles
% near + or - 90.
%
RCUT=R;
for ii=1:20
    RCUT(:,ii)=0;

```

```

    RCUT(:,159+ii)=0;
end
%
% Determine the rotation angle by the max of the Radon matrix.
% Rotate the image and find its new dimensions.
%
YY=max(max(RCUT));
[xpmax,thetamax]=find(RCUT==YY);
rotateangle=-(thetamax-90);
IRC=imrotate(I,rotateangle,'bicubic');
[dimroty,dimrotx]=size(IRC);
%
% Now find the 'vertical profile' of the rotated image.
% We then set a cutwidth of 15% of the vertical size and, ad hoc, cut this profile to keep only that portion of the
image
% within +/- cutwidth/2 rows of the ship's horizontal axis.
%
array=sum(IRC,2);
YIRCMAx=max(array);
rowline=find(array==YIRCMAx);
cutwidth=round(0.15*dimroty);
isodd=mod(cutwidth,2);
if isodd == 1
    cutwidth=cutwidth+1;
end;
upperline=rowline+(cutwidth/2);
if upperline > dimroty
    upperline=dimroty;
end
lowerline=rowline-(cutwidth/2);
if lowerline < 1
    lowerline=1;
end;
CUT=zeros(size(IRC));
for iy=lowerline:upperline
    CUT(iy,:)=IRC(iy,:);
end
%
% Now form the 'profile' of the ship by summing over the rows of the 'cut' image.
%
arrx=sum(CUT,1);
XIRCMAx=max(arrx);
%
% Now approximately center the profile and then cut and 'pad' the image to have a uniform width of 512 pixels (for
% FFT purposes).
%
LL=floor((512-dimrotx)/2);
for ii=1:dimrotx
    profile(LL+ii)=arrx(ii)/XIRCMAx;
    IRCCUT(:,LL+ii)=CUT(:,ii);
end
for ii=1:LL
    profile(ii)=0;
    IRCCUT(:,ii)=0;
    profile(LL+dimrotx+ii)=0;
    IRCCUT(:,LL+dimrotx+ii)=0;
end
profile(512)=0;
IRCCUT(:,512)=0;
%
% Take the FFT of the profile.
% Zero the DC component, normalize the spectrum to 1, and save the values from 2 to specdim+1
% to serve as the image feature vector.
%
spectrum=abs(fft(profile));
fftspectrum=fftshift(spectrum);

```



```

for ii=1:256
    monospectrum(ii)=fftspectrum(ii+256);
end
monospectrum(1)=0;
specmax=max(monospectrum);
normmonospectrum=monospectrum/specmax;
normspec=normmonospectrum(2:specdim+1);
for ii=1:specdim
    fv(ii)=normspec(ii);
end
%
%=====
%
% The next portion of this routine is a mensuration calculation which determines a best estimate of the ship's length
% measured in image pixels. Five different approaches to this calculation are carried out to enable a comparison of
% these approaches.
% Note that this parameter of "bestwidth" was not used in the image feature vector owing to a lack of sufficient data
% needed to determine the ship width in absolute length units.
%
% We now compute the Convolution and Correlation matrices of the ship profile with a step pulse of width =
% stepwidth.
% This leads to the vectors peakcov and peakcorr as functions of the stepwidth from 1 to 512.
%
peakcorr=zeros(1,512);
peakcov=zeros(1,512);
for jj=1:1:512
    step=zeros(1,512);
    stepwidth=jj;
    LL=256-round(stepwidth/2)+1;UL=256+ceil(stepwidth/2);
    step(LL:UL)=1;
    %%% C=conv(profile,step); Same as xcorr
    D=xcorr(profile,step);
    E=xcov(profile,step);
    peakcorr(jj)=max(max(D));
    peakcov(jj)=max(max(E));
end
%
% Finally, we integrate the profile relative to the step pulse of width = pulsewidth 1:512 to give a matrix 'total'.
% The best estimate of the image width = bestwidthA is found by finding the first point where peakcov is a maximum.
% The row of total at row no. = 'bestwidthA' will give the starting column for the step pulse of width = bestwidthA.
%
total=zeros(512,512);
%
% The vector peakcov will give us the 'A' bestwidth estimate for the ship length by the coordinate at which peakcov is
% a
% maximum. The corresponding starting point for the shippulse is found from the large matrix "total(bestwidthA,:)"
% as the
% first point of the maximum.
%
maxpeakcov=max(peakcov);
jjpeakcov=find(peakcov==maxpeakcov);
bestwidthA=jjpeakcov(1);
for jj=1:512
    if (jj+bestwidthA <= 512)
        dummy1=profile(jj:jj+bestwidthA);
        dummy2=0;
    else
        dummy1=profile(jj:512);
        overlap=jj+bestwidthA-512;
        dummy2=0;
    end
    total(bestwidthA,jj)=sum(dummy1)+sum(dummy2);
end
startcurve=total(bestwidthA,:);
maxstartcurve=max(startcurve);

```

```

KK=find(startcurve==maxstartcurve);
startpointA=KK(1);
endpointA=startpointA+bestwidthA;
%
% A second 'B' bestwidth estimate for the ship length is found by the first index into peakcorr at which peakcorr max
% is
% > or = to a value = 0.99 of max(peakcorr). The factor 0.99 is an ad hoc factor.
% The corresponding starting point for the shippulseB is found from the large matrix "total(bestwidthB,:)" as the first
% point
% of the maximum.
%
workingmax=0.99*max(peakcorr);
III=find(peakcorr>=workingmax);
bestwidthB=III(1);
for jj=1:512
    if (jj+bestwidthB <= 512)
        dummy1=profile(jj:jj+bestwidthB);
        dummy2=0;
    else
        dummy1=profile(jj:512);
        overlap=jj+bestwidthB-512;
        dummy2=0;
    end
    total(bestwidthB,jj)=sum(dummy1)+sum(dummy2);
end
KKK=find(total(bestwidthB,:)==max(total(bestwidthB,:)));
startpointB=KKK(1);
endpointB=startpointB+bestwidthB;
%
% For interest, we calculate a third "bestwidth" for comparison purposes.
%
product=peakcorr.*peakcov;
NNN=find(product==max(product));
bestwidthC=NNN(1);
for jj=1:512
    if (jj+bestwidthC <= 512)
        dummy1=profile(jj:jj+bestwidthC);
        dummy2=0;
    else
        dummy1=profile(jj:512);
        overlap=jj+bestwidthC-512;
        dummy2=0;
    end
    total(bestwidthC,jj)=sum(dummy1)+sum(dummy2);
end
KKK=find(total(bestwidthC,:)==max(total(bestwidthC,:)));
startpointC=KKK(1);
endpointC=startpointC+bestwidthC;
%
% Now calculate two additional estimates for 'bestwidth' based upon the peakcorr curve.
% The bestwidthD estimate is that point at which the parameter 'intgradstep' first reaches its maximum value.
% Gradstep is a 1x512 vector which is = 1 if the gradient of peakcorr is greater than 0.1 (ad hoc number) and = 0
% otherwise. Intgradstep then is the quantity sum(gradstep(1:ii)).
% The bestwidthE estimate is an attempt to improve on bestwidth D by allowing for the case where the peakcorr
% curve
% shows a 'flat' region near the final 'flat' region, i.e., where there is a near-zero gradient followed by a short increase
% to
% the final near-zero gradient. It does this by defining the 'edgepoint' to be the last point in gradstep which is = 1
% AND
% which has at least 4 of the nearest points to it also equal to 1 (including, of course, the point itself).
%
deriv=diff(peakcorr);
deriv=deriv./(max(deriv));
grad=gradient(peakcorr);
grad=grad./(max(grad));
gradstep=zeros(1,512);

```



```

for ii=1:512
    if grad(ii) > 0.1
        gradstep(ii)=1;
    end
    intgradstep(ii)=sum(gradstep(1:ii));
end
vetgradstep=gradstep;
edgepoint=zeros(1,512);
for ii=10:500
    points=sum(gradstep(ii-2:ii+2));
    if gradstep(ii)==1
        if points >= 4
            edgepoint(ii)=1;
        end
    end
end
edgepoint(1:9)=1;
[CC]=find(edgepoint==1);
maxintgrad=max(intgradstep);
[BB]=find(intgradstep==maxintgrad);
bestwidthD=BB(1);
bestwidthE=max(CC);
for jj=1:512
    if (jj+bestwidthD <= 512)
        dummy1=profile(jj:jj+bestwidthD);
        dummy2=0;
    else
        dummy1=profile(jj:512);
        overlap=jj+bestwidthD-512;
        dummy2=0;
    end
    total(bestwidthD,jj)=sum(dummy1)+sum(dummy2);
end
for jj=1:512
    if (jj+bestwidthE <= 512)
        dummy1=profile(jj:jj+bestwidthE);
        dummy2=0;
    else
        dummy1=profile(jj:512);
        overlap=jj+bestwidthE-512;
        dummy2=0;
    end
    total(bestwidthE,jj)=sum(dummy1)+sum(dummy2);
end
KKK=find(total(bestwidthD,:)==max(total(bestwidthD,:)));
startpointD=KKK(1);
endpointD=startpointD+bestwidthD;
KKK=find(total(bestwidthE,:)==max(total(bestwidthE,:)));
startpointE=KKK(1);
endpointE=startpointE+bestwidthE;
%
% Now calculate two "zero profiles" with the ship pulse subtracted to give a profile of zero average. Also calculate its
% FFT.
%
average=(sum(profile(startpointA:startpointA+bestwidthA)))/bestwidthA;
shippulseA=zeros(1,512);
shippulseA(startpointA:startpointA+bestwidthA)=average;
zprofileA=zeros(1,512);
zprofileA=profile-shippulseA;
%
average=(sum(profile(startpointB:startpointB+bestwidthB)))/bestwidthB;
shippulseB=zeros(1,512);
shippulseB(startpointB:startpointB+bestwidthB)=average;
zprofileB=zeros(1,512);
zprofileB=profile-shippulseB;
%

```

```

average=(sum(profile(startpointC:startpointC+bestwidthC)))/bestwidthC;
shippulseC=zeros(1,512);
shippulseC(startpointC:startpointC+bestwidthC)=average;
zprofileC=zeros(1,512);
zprofileC=profile-shippulseC;
%
average=(sum(profile(startpointD:startpointD+bestwidthD)))/bestwidthD;
shippulseD=zeros(1,512);
shippulseD(startpointD:startpointD+bestwidthD)=average;
zprofileD=zeros(1,512);
zprofileD=profile-shippulseD;
%
average=(sum(profile(startpointE:startpointE+bestwidthE)))/bestwidthE;
shippulseE=zeros(1,512);
shippulseE(startpointE:startpointE+bestwidthE)=average;
zprofileE=zeros(1,512);
zprofileE=profile-shippulseE;
%
setfft=fft(profile);
% Note that 2=512, 3=511, 4=510, ....., 256=258, 257=257
cutoff=64;
upcut=514-cutoff;
setfft(cutoff:upcut)=0;
calmer=ifft(setfft);
modspectrum=abs(fft(zprofileE));
shiftedmodspectrum=fftshift(modspectrum);
for ii=1:256
    monomodspectrum(ii)=shiftedmodspectrum(ii+256);
end
%
%=====
%
%
% Now display various plots of the image and data results from above.
%
figure;
captionb=sprintf(' %i Rows by %i Columns',N(2),N(1));
precaptionc=sprintf('Unfiltered Image : %s',longfilename);
captionc=strrep(precaptionc,' ','\ ');
orient landscape;
imagesc(A);axis equal;axis tight;axis xy;colormap(jet);zoom on;
caption = strcat(captionc,captionb);
title(caption);
%
%
figure;
captionb=sprintf(' %i Rows by %i Columns',N(2),N(1));
precaptionc=sprintf('Filtered Image : %s',longfilename);
captionc=strrep(precaptionc,' ','\ ');
orient landscape;
imagesc(I);axis equal;axis tight;axis xy;colormap(jet);zoom on;
caption = strcat(captionc,captionb);
title(caption);
%
%
figure;
captionb=sprintf(' %i Rows by %i Columns',N(2),N(1));
precaptionc=sprintf('Edge Detection on Image : %s',longfilename);
captionc=strrep(precaptionc,' ','\ ');
orient landscape;
imagesc(BW);axis tight;axis xy;zoom on;colormap(gray);
caption = strcat(captionc,captionb);
title(caption);
%
%
figure;

```



```

[NR,NC]=size(R);
captionb=sprintf(' %i Rows by %i Columns',NR,NC);
precaptionc=sprintf('Radon Transform of Image : %s',longfilename);
captionc=strrep(precaptionc,'_','\ ');
orient landscape;
imagesc(theta,xp,R);axis tight;axis xy;colormap(hot);colorbar;zoom on;
xlabel('\theta (degrees)');ylabel('x\prime');
caption = strcat(captionc,captionb);
title(caption);

%
%
figure;
[NR,NC]=size(RCUT);
captionb=sprintf(' %i Rows by %i Columns',NR,NC);
precaptionc=sprintf('Clipped Radon Transform of Image : %s',longfilename);
captionc=strrep(precaptionc,'_','\ ');
orient landscape;
imagesc(theta,xp,RCUT);axis tight;axis xy;colormap(hot);colorbar;zoom on;
xlabel('\theta (degrees)');ylabel('x\prime');
caption = strcat(captionc,captionb);
title(caption);

%
% Now display the rotated, filtered image.
%
figure;
[NR,NC]=size(IRC);
captionb=sprintf(' %i Rows by %i Columns',NR,NC);
captionc=sprintf('Image : %s Rotated %i degrees',longfilename,-rotateangle);
orient landscape;
imagesc(IRC);axis equal;axis tight;axis xy;zoom on;
caption = strcat(captionc,captionb);
title(caption);

%
%
figure;
[NR,NC]=size(CUT);
captionb=sprintf(' %i Rows by %i Columns',NR,NC);
captionc=sprintf('Cropped Image : %s Rotated %i degrees',longfilename,-rotateangle);
orient landscape;
imagesc(CUT);axis equal;axis tight;axis xy;zoom on;
caption = strcat(captionc,captionb);
title(caption);

%
%
figure;
orient landscape;
plot(array/YIRCMAx,'b-');axis tight;zoom on;hold on;
yline=[0,1];xline=[rowline,rowline];
xlineupper=[upperline,upperline];xlinelower=[lowerline,lowerline];
HL=line(xline,yline,'Color','r');
line(xlinelower,yline,'Color','r');line(xlineupper,yline,'Color','r');
caption=sprintf('Vertical Profile with a Cutwidth = %3i',cutwidth);
title(caption);

%
%
%%%figure;
%%% orient landscape;
%%% plot(arrx/XIRCMAx,'r-');axis tight;zoom on;
%%% title('Horizontal Profile');

%
%
figure;
orient landscape;
plot(profile,'r-');axis tight;zoom on;
title('Horizontal Profile 512');

%

```

```

%%
%% figure;
%% orient landscape;
%% plot(fftspectrum,'r-');axis tight;zoom on;
%% title('Shifted FFT Spectrum of Horizontal Profile 512');
%
%
figure;
orient landscape;
plot(monospectrum,'r-');axis tight;zoom on;
title('One-Sided FFT Spectrum of Horizontal Profile 512');
%
%
figure;
plot(peakcorr,'b-');hold on;plot(peakcov,'r-');axis tight;
legend('Correlation','Convolution');zoom on;
%
%
%% figure;
%% contourf(total,10);axis tight;colorbar;
%
%
%% figure;
%% plot(tops,'b-');axis tight;
%
%
figure;
plot(profile,'b-');hold on;plot(shippulseA,'r-');plot(shippulseB,'g-');
plot(shippulseC,'m-');plot(shippulseD,'c-');plot(shippulseE,'k-');axis tight;
tstringA=sprintf('BestWidthA = %3i',bestwidthA);
tstringB=sprintf('BestWidthB = %3i',bestwidthB);
tstringC=sprintf('BestWidthC = %3i',bestwidthC);
tstringD=sprintf('BestWidthD = %3i',bestwidthD);
tstringE=sprintf('BestWidthE = %3i',bestwidthE);
text(20,0.90,tstringA);
text(20,0.84,tstringB);
text(20,0.78,tstringC);
text(20,0.72,tstringD);
text(20,0.66,tstringE);
legend('Profile','BestwidthA','BestwidthB','BestwidthC','BestwidthD','BestwidthE');zoom on;
%
%
figure;
subplot(3,1,1);plot(profile,'b-');hold on;plot(shippulseA,'r-');plot(shippulseE,'g-');axis tight;zoom on;
subplot(3,1,2);plot(zprofileA,'b-');hold on;plot(shippulseA,'r-');axis tight;zoom on;
subplot(3,1,3);plot(zprofileE,'b-');hold on;plot(shippulseE,'g-');axis tight;zoom on;
%
%
figure;
[ NR,NC]=size(CUT);
captionb=sprintf(' %i Rows by %i Columns',NR,NC);
captionc=sprintf('Cropped Image : %s Rotated %i degrees',longfilename,-rotateangle);
orient landscape;
imagesc(IRCCUT);axis equal;axis tight;axis xy;
caption = strcat(captionc,captionb);
title(caption);
xline=[rowline,rowline];yline=[0,dimroty];
xlineupper=[startpointA+bestwidthA,startpointA+bestwidthA];xlinelower=[startpointA,startpointA];
line(xlinelower,yline,'Color','r');line(xlineupper,yline,'Color','r');zoom on;
xline=[rowline,rowline];yline=[0,dimroty];
xlineupper=[startpointB+bestwidthB,startpointB+bestwidthB];xlinelower=[startpointB,startpointB];
line(xlinelower,yline,'Color','g');line(xlineupper,yline,'Color','g');
xlineupper=[startpointC+bestwidthC,startpointC+bestwidthC];xlinelower=[startpointC,startpointC];
line(xlinelower,yline,'Color','m');line(xlineupper,yline,'Color','m');
xlineupper=[startpointD+bestwidthD,startpointD+bestwidthD];xlinelower=[startpointD,startpointD];
line(xlinelower,yline,'Color','c');line(xlineupper,yline,'Color','c');

```



```

xlineupper=[startpointE+bestwidthE,startpointE+bestwidthE];xlinelower=[startpointE,startpointE];
line(xlinelower,yline,'Color','y');line(xlineupper,yline,'Color','y');
zoom on;
%
%
figure;
plot(monospectrum,'r-');axis tight;zoom on;
%% plot(monomodpectrum,'r-');axis tight;zoom on;
caption='Fourier Spatial Spectrum with DC Suppression';
title(caption);
%
figure;
plot(fv,'r-');axis tight;zoom on;
caption='Normalized Feature Vector';
title(caption);
%
figure;
subplot(2,1,1);plot(profile,'b-');hold on;plot(shippulseE,'r-');axis tight;zoom on;
%% plot(shippulseB,'g-');axis tight;zoom on;
subplot(2,1,2);plot(zprofile,'b-');hold on;plot(shippulseE,'r-');axis tight;zoom on;
%% plot(shippulseB,'g-');axis tight;zoom on;
%
%
figure;
%% subplot(2,1,1);plot(profile,'b-');axis tight;zoom on;
%% subplot(2,1,2);plot(abs(calmer),'b-');axis tight;zoom on;
plot(profile,'b-');hold on;plot(abs(calmer),'r-');axis tight;zoom on;
legend('Original Profile','FFT Filtered Profile');
%
%
%% figure;
%% for ii=1:511
%% diffcorr(ii)=peakcorr(ii+1)-peakcorr(ii);
%% end
%% plot(diffcorr,'r-');axis tight;zoom on;
%
%
figure;
plot(product,'r-');axis tight;zoom on;
%
%
%% figure;
%% iistart=bestwidthA-30;iiend=bestwidthA+140;
%% for ii=iistart:10:iiend
%% plot(total(ii,:), 'r-');hold on;
%% end
%% axis tight;
%
%
figure;
plot(0.99*gradstep,'b-');hold on;plot(intgradstep./maxintgrad,'r-');axis tight;
%
%
%% fprintf('\nBestwidthA = %3i StartPointA = %3i EndPointA = %3i',bestwidthA,startpointA,endpointA);
%% fprintf('\nBestwidthB = %3i StartPointB = %3i EndPointB = %3i',bestwidthB,startpointB,endpointB);
%% fprintf('\nBestwidthC = %3i StartPointC = %3i EndPointC = %3i',bestwidthC,startpointC,endpointC);
%% fprintf('\nBestwidthD = %3i StartPointD = %3i EndPointD = %3i',bestwidthD,startpointD,endpointD);
%% fprintf('\nBestwidthE = %3i StartPointE = %3i EndPointE = %3i',bestwidthE,startpointE,endpointE);
%% fprintf('\n\n\n');
%
%=====
===

```

This page intentionally left blank

[illegible]

INDUSTRY CANADA / INDUSTRIE CANADA

208976

