

ANDROID BASED SHIP RECOGNITION SYSTEM

POOVIZHI.M

10PCA16

**A Project Report submitted to Avinashilingam Deemed University for
Women, Coimbatore in partial fulfillment of the requirements for the
Master's Degree in Computer Applications**

MAY-2013

**SIGNATURE OF THE
HEAD OF THE DEPARTMENT**

**SIGNATURE OF THE
SUPERVISOR**

**SIGNATURE OF THE
EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I would like to express my sincere thanks to God Almighty, for his constant love and grace that he has showered upon me.

I am very grateful to **Dr.T.S.K.Meenakshi Sundaram, M.A., M.Phil., Ph.D.,** Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for his support and encouragement during the course of my study.

I heartily thank **Dr.(Mrs.)Sheela Ramachandran M.Sc., P.G.Dip., Ph.D.,** Vice Chancellor Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for extending all resources that facilitated the conduct of the present study.

I express my humble gratitude to **Dr.(Mrs.) Gowri Ramakrishnan M.Sc., M.Phil., Ph.D.,** Registrar Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing all facilities necessary for the study.

I am also thankful to **Dr.(Mrs.) R. Parvatham M.Sc., Dip.Ed. M.Phil., Ph.D.,** Dean Faculty of Science, for granting the facility required.

I wish to place on record my deep sense of gratitude to **Dr.(Mrs.)G.Padmavathi M.Sc., M.Phil., Ph.D.,** Professor and Head, Department of Computer Science, for providing all the facilities to complete the project.

I take this unique opportunity to express my sincere thanks to my project Coordinator **Dr.(Mrs.)V.Srividhya, M.Sc., M.Phil., Ph.D., Assistant Professor, Department of Computer Science,** for her kind advice and knowledgeable suggestion, which helped me to complete my project successfully.

I owe great deal of gratitude to my esteemed guide **Dr. (Mrs.) P.Subashini, M.C.A., M.Phil., Ph.D., Professor,** Department of Computer Science, for her guidance, imparting the tremendous assistance and knowledgeable suggestion, which helped me to complete my project successfully.

I am greatly indebted to respected madam **Dr.Ms.M.Krishnaveni, M.Sc., M.Phil., Ph.D.,** Department of Computer Science Research Assistant for her inspiring and constant

encouragement and timely guidelines at every stage of this project and who helped me to complete this project a successful one.

I express my humble gratitude to **Dr. (Mrs.)Saroja Prabhakaran, M.A,Dip.Ed.,Ph.D., Director, Hall of Residence**, for her support in completion of the project.

I have great pleasure in expressing my deep sense of gratitude to all other staffs and non-teaching staffs who stood behind the screen in making of project.

I would extend my hearty thanks to one and all that helped me directly or indirectly for successful completion of my project.

Last yet importantly, I would like to thank my parents, my grandfather, my brother and all my well-wishers for their kind inspiration.

SYNOPSIS

The project entitled “**Android based ship recognition system**” is used to detect and recognize the ship based on android software in SAR Image. Satellite-based Synthetic Aperture Radar (SAR) provides a powerful vessel surveillance capability. Due to the presence of speckle noise and to the reduced dimensions of the targets compared to the sensor resolution, the automatic interpretation of SAR images is often complicated even though vessels undetected are sometimes visible by eye. Ship recognition from remote sensing plays a very important role in ship monitoring due to some of its virtues such as a long operating distance and a wide monitoring range. Ship detection in synthetic aperture radar ocean imagery is used to detect the bright targets against the ocean clutter background.

The SAR image processing and target recognition become a hot issue in the field of marine application. Android is very flexible and provides many tools for developing applications. The purpose of the system is basically for surveillance of the ships in marine. The detection and recognition of the target is accomplished by applying image processing techniques such as Segmentation, Feature extraction, Classification and Recognition in android platform. This system mainly helps for detection and recognition through mobile and making such a system is suitable for real-time applications. Such a system helps to recognize different types of ships such as war ship, passenger ship, tanker ship, boat, etc.

CONTENTS

	Page No
1. INTRODUCTION	1
1.1 ABOUT THE SYSTEM	1
1.2 OVERVIEW OF THE PROJECT	2
2. SYSTEM CONFIGURATION	3
2.1 HARDWARE SPECIFICATION	3
2.2 SOFTWARE SPECIFICATION	3
2.3 ABOUT THE SOFTWARE	4
3. SYSTEM STUDY AND ANALYSIS	8
3.1 EXISTING SYSTEM	8
3.2 PROPOSED SYSTEM	8
4. SYSTEM DESIGN	9
4.1 DATASET USED	9
4.2 INPUT DESIGN	10
4.3 OUTPUT DESIGN	10

5. SYSTEM DEVELOPMENT	11
5.1 PREPROCESSING	11
5.2 SEGMENTATION	13
5.3 FEATURE EXTRACTION	14
5.4 CLASSIFICATION AND RECOGNITION OF SHIP	16
6. RESULTS AND DISCUSSION	18
7. CONCLUSION	19
8. SCOPE FOR FUTURE ENHANCEMENT	20
BIBLIOGRAPHY	21
APPENDIX	
9.1 SYSTEM FLOW DIAGRAM	
9.2 DATASET USED	
9.3 INPUT SCREENS	
9.4 OUTPUT SCREENS	
9.5 GUI SNAPSHOTS	
9.6 RESULTS	
9.6.1 INPUT SAR IMAGES AND PREPROCESSED IMAGES	
9.6.2 SEGMENTATION	
9.6.3 FEATURE EXTRACTION	
9.6.4 CLASSIFICATION AND RECOGNITION OF SHIPS	

1. INTRODUCTION

1.1 ABOUT THE SYSTEM

This section is used to describe the problem definition, overview of the project and the purpose for which the project is developed and used.

Android based ship recognition system is used to detect and recognize the ship based on android software in SAR Image. The detected and recognition of the ship is done by the threshold based Segmentation and Back Propagation Neural Network.

Ship recognition from SAR image is a process using Image processing and Neural Network techniques by MATCLIPSE as a software tool. The open-source MATCLIPSE project is an interface between the Java integrated development environment (IDE) Eclipse and Matlab.

The aim is to develop a system for the ship recognition from SAR image. It is done by using image processing steps such as preprocessing, segmentation, feature extraction, classification and recognition.

MAIN OBJECTIVES

The objectives of ship recognition is

- To develop a ship recognition system in android platform.
- To develop an Image processing based applications in android platform.
- To identify the ship in SAR image using back propagation neural network.

1.2 OVERVIEW OF PROJECT

The android based ship recognition system is carried out by image processing technique from Matclipse in Android platform. The Ship detection and identification process in SAR images includes the following steps

- Preprocessing
- Segmentation
- Feature Extraction
- Classification and Recognition

Preprocessing consist of two steps, noise removal and edge detection. Median filter is used to remove the speckle noise in the SAR image. Speckle noise is a granular noise that inherently exists in and degrades the quality of the active radar and synthetic aperture radar (SAR) images. To detect the edge in the image canny operator is used.

The threshold segmentation method is applied for the SAR ship target images to form a separate ship target images. Then for the feature extraction, segmented ship target images and threshold value are taken as input and the output of the images is features such as area, centroid, perimeter, mean, median and variance. The extracted features are taken as input for the classification method. The classifications and recognition of ship are carried out by the back propagation neural network classifier. The back propagation neural network consists of three layers namely input layer, hidden layer and output layer. In the input layer, the extracted features are assigned, hidden layer consists of hidden neurons and the output layer consists of target vectors. The back propagation neural network classifier is trained and tested according to the specified input layer and target layer. The trained and tested data are plotted in the graph. Finally, the proposed system is designed using GUI. It provides a set of tools for creating graphical user interface to greatly simplify the process of designing. This system is done using MATCLIPSE software.

2. SYSTEM CONFIGURATION

This section describes the hardware and software specification needed for both development and implementation phases of this project.

2.1 HARDWARE SPECIFICATION

Processor : Intel® Core(TM) i3-2310M CPU @
2.10GHz 2.10 GHz

RAM : 4.00GB

System type : 64 –bit Operating System

Keyboard : Standard PS/2 key board

2.2 SOFTWARE SPECIFICATION

Software : Android SDK, Java JDK, Eclipse IDE,
Matclipse

Operating System : Microsoft Windows 7

2.3 ABOUT THE SOFTWARE

ANDROID SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices .

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

ECLIPSE IDE

Eclipse as an integrated development environment (IDE) for Java. Today it is the leading development environment for Java with a market share of approximately 65%. Eclipse is created by an Open Source community and is used in several different areas, e.g. as a development environment for Java or Android applications. The Eclipse Open Source community has over 200 Open Source projects covering different aspects of software development.

The Eclipse projects are governed by the Eclipse Foundation. The Eclipse Foundation is a non-profit, member supported corporation that hosts the Eclipse Open Source projects and helps to cultivate both an Open Source community and an ecosystem of complementary products and services. The Eclipse IDE can be extended with additional software components. Eclipse calls this software components plug-ins. Several Open Source projects and companies have extended the Eclipse IDE.

MATCLIPSE

The open-source matchclipse project is an interface between the Java integrated development environment (IDE) Eclipse and Matlab. It was developed at the Institute for Theoretical and Computational Physics (ITP) at Graz University of Technology (TU Graz) by a team that consisted of David Camhy, Winfried Kernbichler, Georg Huhs and Christopher Albert. Matchclipse is part of a bigger endeavor, namely, the development of the teaching and learning environment MatlabTutor. It is released under the Eclipse Public License (full text).

The aim of the project is to facilitate the usage of Matlab directly from Eclipse under various operating systems (Linux, Windows, OS X). At the moment, it consists of a workbench, a Matlab editor, a Matlab console including a command window and a viewer for results, a Matlab command history and a Matlab workspace viewer.

Matlab Workbench

The Matlab workbench uses a tree structure for workspace. One can create Matlab projects there and, of course, all basic features from Eclipse are supported there. Projects can have a Matlab nature and files can be executed directly from the workbench without opening them in the editor.

Communication with Matlab

At the moment, the communication between Java and Matlab is based on the usage of pipes under Linux and OS X and on the usage of DCOM under Windows. In both cases it is necessary to have a local installation of Matlab.

In Linux as well as OS X there is no documented way of access to Matlab, to its memory and to its variables. For that reason, the most reliable way of communication was the usage of pipes. Since in this case, memory in Matlab and memory in Eclipse is strictly separated, any transfer of variables from Matlab to Eclipse needs time and doubles the amount of necessary memory. Therefore, this is not the best of all ways but without the help of MathWorks no other possibility is feasible. Under the operating system Windows, MathWorks provides DCOM as a possibility to control Matlab.

Additional problems arises if graphical output is produced in Matlab or if Matlab tools like the help browser or the profile viewer has to be used. One way to handle this problem is the installation of an X11 server on the client which could display all additional

windows from the server on the client through the X11 protocol in case of a correct setting of the display variable. Beside issues of security and speed, technically everything could run as on a local machine but the requirement is a reliable X11server used by Eclipse on the client side.

Matlab Editor

The Matlab editor is based on a regular expression parser which provides syntax highlighting and checking. A new Xtext-based version had been under development but was not been brought to a usable state due to the limitations of the features and performance of the Xtext framework.

The current features are:

- Smart indentation and parenthesis matching.
- Detection of variables, operators, strings, commands and key words.
- Integration of the Matlab program mlint to mark errors and warnings directly in the editor.

Matlab Console

In contrast to Matlab where input to and output from Matlab is displayed in the same window, *matclipse* uses a concept with a separate command line for input to Matlab and a results viewer for displaying the output from Matlab nicely separated by the input commands. This concept proved to be practical but has its limitations.

The current features of the command line are the following:

- Syntax highlighting is provided by the editor.
- Up- and down-scrolling in the command history.
- One can start the Matlab help browser from an icon.
- One can clear the results view from an icon (*clc* does not work).
- One can toggle Matlab console debug from an icon, to debug the XML communication between *matclipse* and Matlab. Note that this debug mode does not enable debugging the Matlab code itself – perhaps this will be added in the future.

3. SYSTEM STUDY AND ANALYSIS

System development can generally be thought of having two major components: system analysis and systems design. In system Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desired or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system.

3.1 Existing System

There is an existing automated ship recognition system present for the detection and recognition of SAR ship. This project is android based ship recognition system to detect and recognize the ship on real time basis.

3.2 Proposed System

In the proposed system, android based ship recognition system from SAR images, ships in the images are detected using threshold based segmentation and target ship are recognized by the back propagation neural network with the help of an extracted feature. As the result, SAR ship is identified and recognized in android platform.

Merits of proposed system

- Helps in marine monitoring.
- Easy access of application in mobile device.
- Potential for free software.

4. SYSTEM DESIGN

Design is the first process in the development phase of any engineered system. The inputs to the design are the software requirements and the output will be the design specification applicable to all software.

4.1 DATASET

In this project, SAR ship images and extracted feature vectors are considered as the dataset since the project deals with ship classification and recognition from SAR image. They are shown in APPENDIX [9.2.1, 9.2.3]. Ship recognition system is capable to detect and recognize the SAR ship target according to the extracted feature vectors. The feature vectors are extracted from the segmented SAR ship images. The ship recognition system performs training and testing data from the extracted feature vectors and the classification is done according to their correspond classes. In ship recognition system SAR images are taken as input for pre-processing and segmentation. For features extraction segmented SAR image with threshold values is used and feature vectors dataset are taken as input for back propagation neural network classifier and they are shown in APPENDIX [9.2.2, 9.2.3].

4.1.1 Image Formats

Most popular image types are,

- ✓ BMP (.bmp) - Microsoft Windows Bitmap Format
- ✓ GIF (.gif) - Graphic Interchange Format
- ✓ JPEG (.jpeg) - Joint Photographic Experts Group Format
- ✓ PNG (.png) - Portable Network Graphics Format
- ✓ TIFF (.tiff) - Tag Image File Format

In the project ship recognition, the images are of type GIF, BMP and JPEG are used for pre-processing, segmentation and features extraction process. Images are shown in the APPENDIX [9.2.1 and 9.2.2].

4.2 INPUT DESIGN

The main objective of the input design is to provide user friendly interaction. The user has to make a minimum, input as a whole process is automated.

As the project is concentrated fully on the android based ship recognition system in SAR image its input are dataset extracted from the SAR images. Input form is designed by using Matclipse platform in GUI. The input designs are shown in APPENDIX [9.3.1]. The input SAR ship images are taken for pre-processing. In pre-processing, noises are removed by median filter and then edge detection is done by canny operator. The pre-processed images are taken as input for the threshold based segmentation method.

The segmented images and threshold value are taken as input for the feature extraction technique. The segmented images are shown in APPENDIX [9.2.2]. Area, perimeter, centroid, mean, median and variance are the features extracted. The feature vector values are shown in APPENDIX [9.2.3]. By using back propagation neural network classifier, the extracted features are trained and tested with the corresponding values. Finally the SAR ships are classified and recognized in SAR image.

4.3 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system. The images and dataset are taken as input. In this project, edge detected SAR images are generated as an output for pre-processing technique. The output of the preprocessing are shown in APPENDIX [9.4.1] Segmented SAR ship target images are extracted as an output for Threshold based Segmentation. The segmented images are shown in APPENDIX [9.4.2]. Then feature vectors of segmented ship target images are extracted. The table is shown in APPENDIX [9.4.3]. The extracted feature vectors are trained and tested with the dataset using back propagation neural network classifier. The tested and trained datasets are plotted corresponding to their classes. The image are shown in APPENDIX [9.4.4].

5. SYSTEM DEVELOPMENT

System development is a series of operations performed to manipulate data to produce output from a computer system. The principle activities performed during the development phase can be divided into a major related sequence. They are

- Internal
- External

The major internal system development activities done for the system is computer program development and performance testing.

The major external system development activities done are planning and implementation.

Module Description

The project consist of four main modules namely

1. Pre-processing
2. Segmentation
3. Feature Extraction
4. Classification and recognition of ship

5.1 PREPROCESSING

In android based ship recognition system, preprocessing process is divided into two steps. They are

- Noise Removal
- Edge Detection

5.1.1 Noise Removal

Reduction of speckle noise is one of the most important processes to increase the quality of Synthetic Aperture Radar (SAR) images. Speckle is a granular noise that inherently exists in and degrades the quality of the SAR images. Filtering is one of the common method which is used to reduce the speckle noises.

Median filtering is a nonlinear operation often used in image processing to reduce noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges. In Median filtering, the value of an output pixel is determined by the median of the neighbourhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values. Median filtering is therefore better able to remove this outlier without reducing the sharpness of the image.

In the project, SAR image is taken as input and median filter is used to remove the speckle noise in the SAR image. Because most of the noise in SAR images are speckle noise. Noise removed image is taken for the edge detection process. Noise removed images are shown in APPENDIX [9.6.1].

5.1.2. Edge Detection

Edges in images are areas with strong intensity contrasts a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image.

The Canny edge detection algorithm is known to many as the optimal edge detector. The canny edge detector smooths the image to eliminate the noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum. The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero. If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from one pixel to another pixel with above gradient level.

In the project, denoised SAR image is taken as input and then edge detection is done by canny operator. The output of pre-processing is an edge detected SAR image. Edge detected images are shown in APPENDIX [9.6.1].

5.2 SEGMENTATION

Image segmentation is an important technology for image processing. There are many applications whether on synthesis of the objects or computer graphic images require precise segmentation. Before analyzing and extracting feature from the object, the image has to be subdivided into constituent parts or objects. The process of sub dividing the images into its constituent parts or objects is called image segmentation.

Segmentation algorithm can be divided into two categories based on the two properties such as,

- Discontinuity
- Similarity

First category is to partition an image based on abrupt changes in intensity, such as edges in an image. Second category is based on partitioning an image into regions that are similar according to predefined criteria. Threshold based segmentation method come under similarity.

Threshold is one of the widely used methods for image segmentation. Threshold techniques can be categorized into two classes: global threshold and local (adaptive) threshold. Here global threshold method is applied for the segmentation. In the global threshold, a single threshold value is used in the whole SAR image. It is useful in discriminating foreground from the background. By selecting an adequate threshold value T , the gray level image can be converted to binary image. The binary image should contain all of the essential information about the position and shape of the objects of interest (foreground). The advantage of obtaining first a binary image is that it reduces the complexity of the data and simplifies the process of recognition and classification.

In the project, threshold based segmentation is used to detect the SAR image. The output of segmentation method contains both threshold value and segmented image. Both image and value are used for the feature extraction process. The results are shown in APPENDIX [9.6.2].

5.3 FEATURE EXTRACTION

Feature extraction in image processing is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant then the input data will be transformed into a reduced representation set of features also named features vector. Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which over fits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

Features extracted from the SAR ship targets are

- Area
- Perimeter
- Centroid
- Mean
- Median
- Variance

AREA

Area is amount of surface the 2D shapes cover. It's measured in square unit. A mathematically acceptable definition of area is complex. Area is usually measured or defined on a flat surface, also called a Euclidean plane, or on a spherical surface. The surface area is occasionally determined for irregular objects. In the case of extremely complex or esoteric surfaces, the area might be impossible to define or measure.

PERIMETER

Perimeter is the distance around a closed figure and is typically measured in millimetres (mm), centimetres (cm), metres (m) and kilometres (km). The word 'perimeter' is also sometimes used instead of circumference. The formula to find the circumference of a circle is $2 \pi r$.

CENTROID

The centroid or geometric center of a two-dimensional region is, informally, the point at which a cardboard cut-out of the region could be perfectly balanced on the tip of a pencil. The centroid of a plane figure or two-dimensional shape is the arithmetic mean position of all the points in the shape. The definition extends to any object in n-dimensional space: its centroid is the mean position of all the points in all of the coordinate directions. The center of mass is the arithmetic mean of all points weighted by the local density or specific weight. If a physical object has uniform density, then its center of mass is the same as the centroid of its shape.

MEAN

The most common expression for the mean of a statistical distribution with a discrete random variable is the mathematical average of all the terms. To calculate it, add up the values of all the terms and then divide by the number of terms. This expression is also called the arithmetic mean. The mean of a statistical distribution with a continuous random variable, also called the expected value, is obtained by integrating the product of the variable with its probability as defined by the distribution.

MEDIAN

The median of a distribution with a discrete random variable depends on whether the number of terms in the distribution is even or odd. If the number of terms is odd, then the median is the value of the term in the middle. This is the value such that the number of terms having values greater than or equal to it is the same as the number of terms having values less than or equal to it. If the number of terms is even, then the median is the average of the two terms in the middle, such that the number of terms having values greater than or equal to it is the same as the number of terms having values less than or equal to it.

VARIANCE

Variance is a mathematical expectation of the average squared deviations from the mean. Variance measures the variability (volatility) from an average. Volatility is a measure of risk, so this statistic can help determine the risk an investor might take on when purchasing a specific security.

In the project android based ship recognition system from SAR ship images, feature vectors are extracted using features formula like area, centroid, perimeter, etc. With this feature vector values, ships in SAR images are classified. The table for the feature vector values are shown in APPENDIX [9.6.3].

5.4 CLASSIFICATION OF VESSELS

In android based ship recognition system, classification classify different class according to the feature vectors extracted from the SAR ship target images. Classification is needed to classify different target according to their specified classes. In this project, back propagation neural network classifier is performed for classification of the ship target. The back propagation neural network classifier is preformed for training and testing the dataset.

The BPN classifier consists of two paths; forward path and backward path. Forward path contain creating a feed forward network, initializing weight, simulation and training the network. The network weights and biases are updated in backward path. Feed forward networks often have one or more hidden layers of sigmoid neurons followed by output layer of linear neurons. Multiple layers of neurons with non linear transfer functions allow the network to learn non linear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range -1 to +1

In the project, extracted feature vectors are taken as the input vector for back propagation neural network classifier and their associated classes A, B C specified as target class. Then BPN classifier is trained according to the corresponding class .Finally the output is plotted in the graph corresponding to their classes. The results of classification are shown in APPENDIX [9.6.4].

The dataset used in the project are chosen from SAR ship target. First, the pre-processed images are taken as an input for the segmentation phase and the segmentation is done using threshold based segmentation technique. Next, the samples were divided into three classes, class A is tanker ships and class B is passenger ship and class C boat. The

extracted features are trained with three class A, B and C. Finally the output is tested with 10 images feature vector values. The results are shown in APPENDIX [9.6.4]. The project was done in the matclipse.

6. RESULTS AND DISCUSSION

In the project, android based ship recognition system from SAR images the results are shown in APPENDIX [9.6]. By using median filter, noises are removed and then edge detection process is done. The output of the pre-processing method is edge detected SAR images. Threshold based Segmentation method is used for segmentation. As a result of segmentation, segmented SAR ship is evaluated. The results of segmentation are shown in APPENDIX [9.6.2].

Then the segmented images are taken as input images for features extraction. In this project feature vectors are extracted by Statistical features such as area, perimeter, centroid, mean, median and variance. The results of feature vectors are shown in APPENDIX [9.6.3]. The extracted features are taken as input for the classification and recognition method. The classifications of ships are carried out by the back propagation neural network classifier. The back propagation neural network was trained, the dataset according to their target vector. Then the classifier is tested and the results are plotted in the graph. The trained and tested dataset are plotted in graph are shown in APPENDIX [9.6.4].

7. CONCLUSION

The project “**Android based ship recognition system**” is implemented successfully and it is more useful for military ground surveillance and earth observation. The objective of ship recognition system is to detect and recognize the ship from SAR images. It is more useful for marine system and military surveillance for detecting and tracking the ship.

Android based ship recognition system is basically developed for mobile device. With this system, portability is occurred and use anywhere in marine ocean.

Back propagation neural network classifier classify the ship corresponds to their classes according to their feature vectors extracted from the segmented SAR images. Feature vectors are trained and tested using back propagation neural network classifier and plotted in the graph.

Android based ship recognition system is not only useful for detect and recognition purpose but it is also useful for tracking of ship in the ocean from SAR images.

The project included all the important features of ECLIPSE IDE and MATCLIPSE software. This project is developed with the aim of ship recognition system in android platform.

8. SCOPE FOR FUTURE ENHANCEMENT

Scope is a measure of considering a necessary part of a project. If something is not in scope, then it is not something you want to spend time building. Scope is very important on large projects. But the real problem was often that too much effort was spent on unnecessary things. People especially volunteer coders and builders have a limited amount of free time and energy to work on projects.

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancement can be done to the applications so that the model functions very attractive and useful manner than the present one.

- ↗ Display results in Emulator.
- ↗ Real Time Image Capture.
- ↗ Implemented in more than one classifier.
- ↗ Improves appearance.

9. BIBLIOGRAPHY

1. Iria Estevez-Ayres, Marisol Garcia-Valls, Pablo Basanta-Val and Ivan Fernandez-Pacheco,2011,” **Using Android Smartphones in a Service-Oriented Video Surveillance System**”, IEEE International Conference on Consumer Electronics (ICCE),pp.887-888.
2. Wang Juan, Sun Lijie , and Zhang Xuelan, 2009,” **Study Evolution of Ship Target Detection and Recognition in SAR Imagery**”, Proceedings of the 2009 International Symposium on Information Processing (ISIP’09) Huangshan, P. R. China, August 21-23, 2009, pp. 147-150 .
3. Chin-Ya Huang, Mon-Ju Wu,2006 ,“**Image Segmentation**”, ECE 533 Final Project, University of Wisconsin-Madison.
4. Salem Saleh Al-amri, N.V. Kalyankar and Khamitkar S.D, 2010,” **Image Segmentation by Using Thershod Techniques**”, JOURNAL OF COMPUTING, VOLUME 2, ISSUE 5, ISSN 2151-9617, pp.83-86.
5. M. Uma Selvi, S. Suresh Kumar, 2011,” **A Novel Approach for Ship Recognition using Shape and Texture**”, International Journal of Advanced Information Technology (IJAIT) Vol. 1, No.5, pp.23-29.
- 6.Paris W. Vachon, 2006,” **Ship Detection in Synthetic Aperture Radar Imagery**”, Proceedings Ocean SAR 2006 – Third Workshop on Coastal and Marine Applications of SAR, St. John’s, NL, Canada.
7. James K.E. Tunaley,” **Ship Detection in SAR Imagery**”, LRDC2010-12-01-001.
8. Leszek Rutkowski, 2004, “**Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment**” IEEE Transactions on Neural Networks, Vol. 15, NO. 4.
9. L. Gagnon and R. Klepko , 1998,” **Hierarchical Classifier Design for Airborne SAR Images of Ships**”, SPIE Proc. #3371.
10. H. Osman, S. Blostein, L. Gagnon, 1997,“**Classification of Ships in Airborne SAR Imagery Using Backpropagation Neural Networks**”, *SPIE Proc. #3161*, pp. 126-136.
11. R. Klepko, , 1995, “**Automatic Pattern Classification of Airborne SAR Images of Ships**”, *DREO Report #1283*, Ottawa.
12. Khairul Muzzammil bin Saipullah, Ammar Anuar, Nurul Atiqah binti Ismail, and Yewguan Soo, 2012,” **Real-Time Video Processing Using Native Programming on**

Android Platform”, IEEE 8th International Colloquium on Signal Processing and its Applications.

13. Michael T. Wells, 2008, “**Mobile Image Processing on the Google Phone with the Android Operating System**”, pp.1-34.

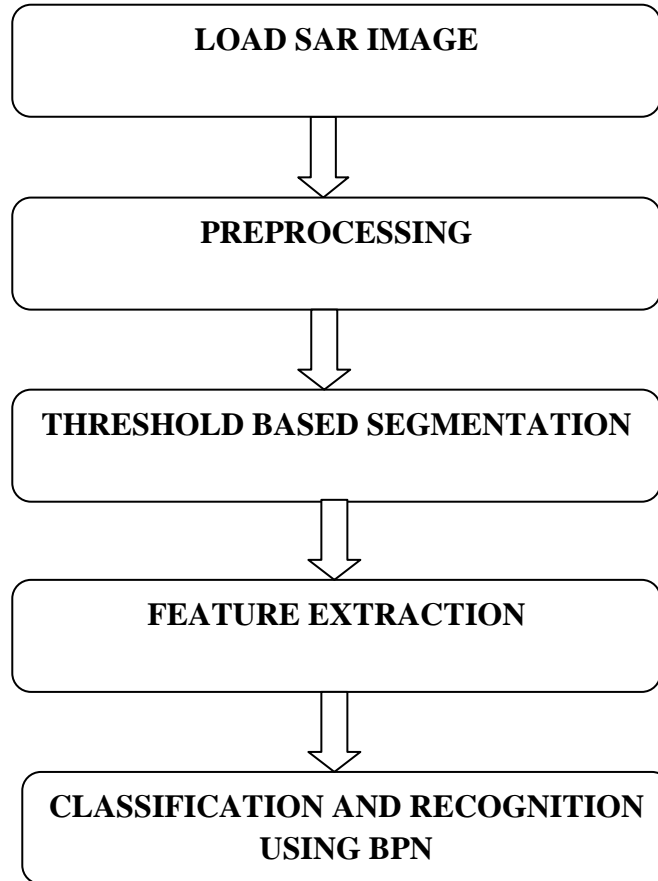
14. Saidi Mohamed nabil, Toumi abdelmalek, Khenchaf ali, Aboutajdine driss, Hoeltzener brigitte, “**Feature extraction and fusion for Automatic target recognition based ISAR images**”, laboratory e3i2-ea3876 ensieta, brest, france.

WEBSITES

1. <http://www.openframeworks.cc/setup/android-eclipse/>
2. <http://mobile.tutsplus.com/tutorials/android/java-tutorial/>
3. <http://apcmag.com/building-a-simple-android-app.htm>
4. <http://coenraets.org/blog/androidtutorial/>
5. <http://www.mkyong.com/android/android-imageview-example/>
6. <http://yournextleap.blogspot.in/2012/02/insert-image-to-database-and-display.html>
7. <http://mobile.tutsplus.com/tutorials/android/android-sdk-creating-forms/>
8. <http://www.tomgibara.com/computer-vision/CannyEdgeDetector.java>
9. <http://code.google.com/p/android-motion-detection/source/list>
10. <http://teyvoniatthomas.com/index.php/projects/56-opencv-on-android.html>
11. <http://www.freedownloadmanager.org/download/android-edge-detection-code-5022892.html>
12. <http://undocumentedmatlab.com/blog/matclipse-eclipse-matlab-interface/>
13. <http://www.slideshare.net/mentelibre/neural-network-toolbox-matlab>
14. <http://homepages.cae.wisc.edu/~ece539/matlab/>
15. http://s.pudn.com/search_hot_en.asp?k=matlab+bpn
16. <http://amath.colorado.edu/computing/Matlab/Tutorial/Programming.html>

APPENDIX

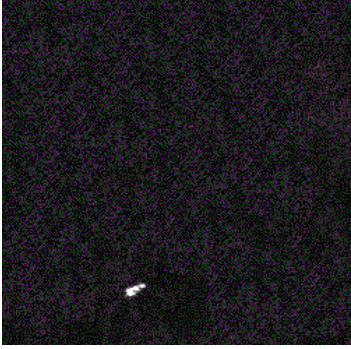
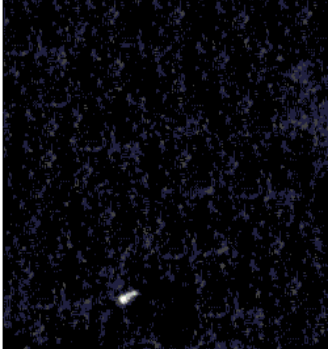
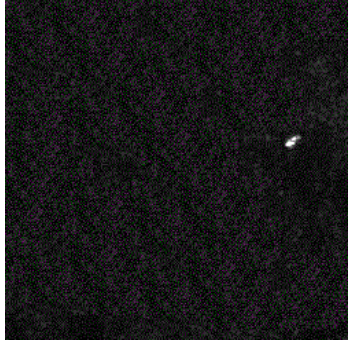
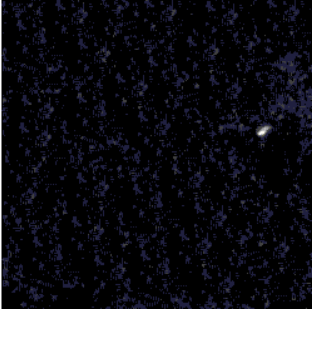
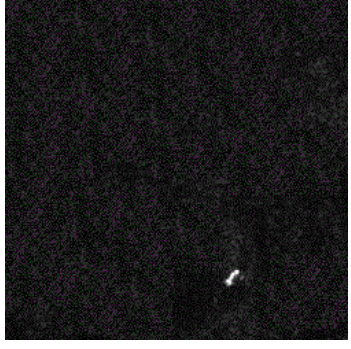
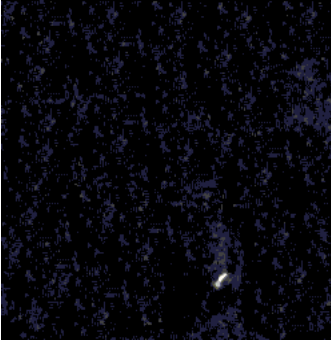
9.1 SYSTEM FLOW DIAGRAM

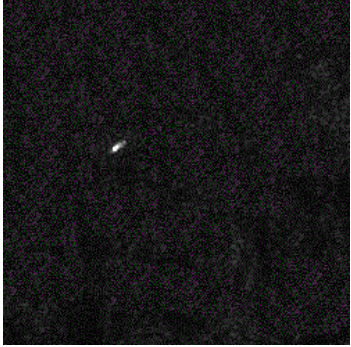
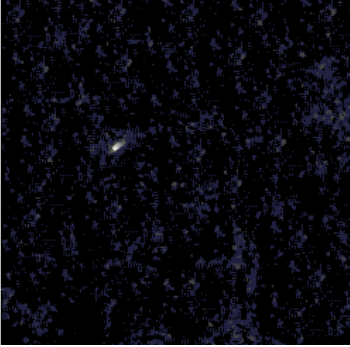
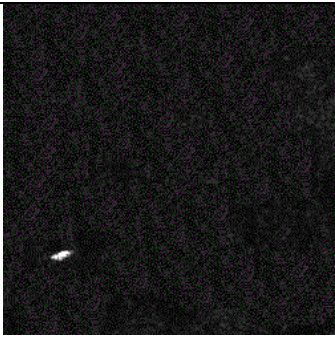
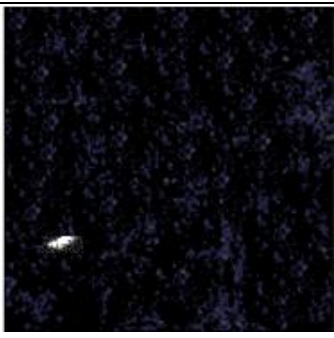
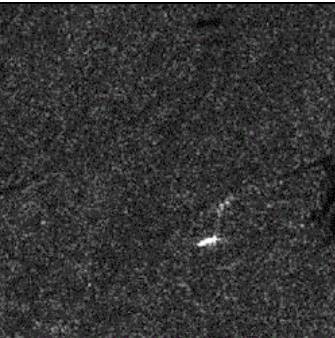
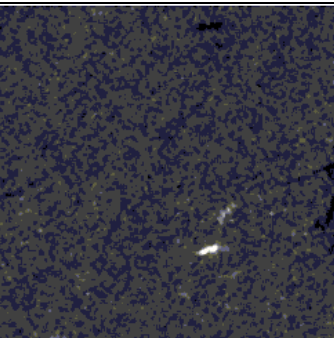

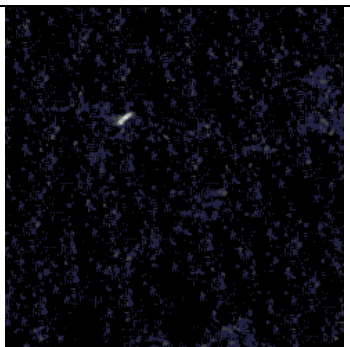


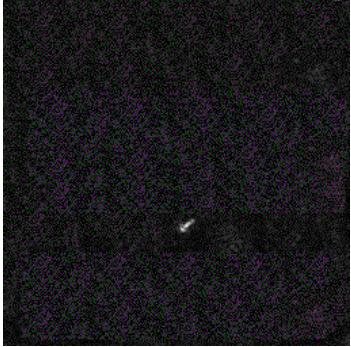
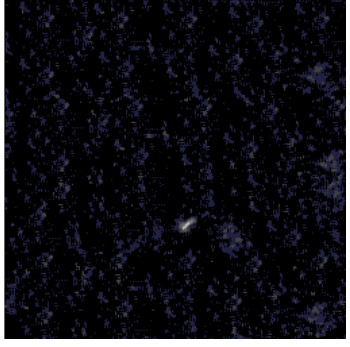
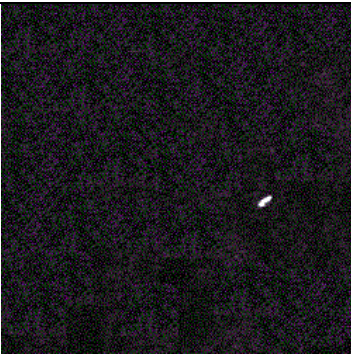
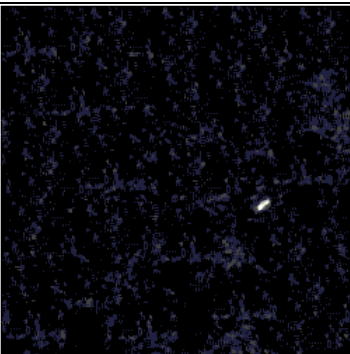
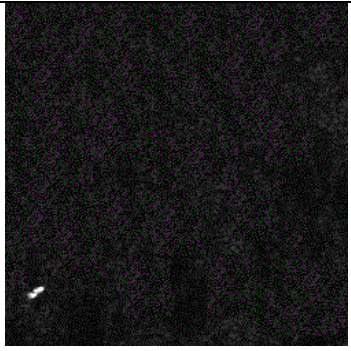
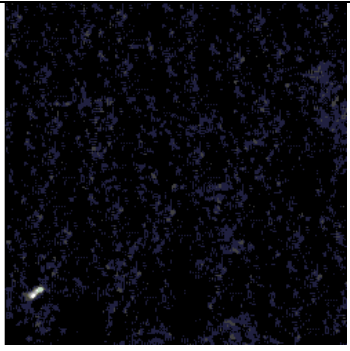
9.2 DATASET USED

9.2.1 DATASET USED FOR PREPROCESSING

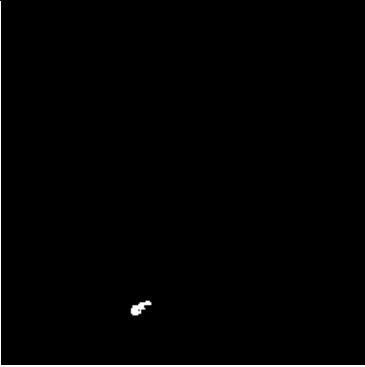
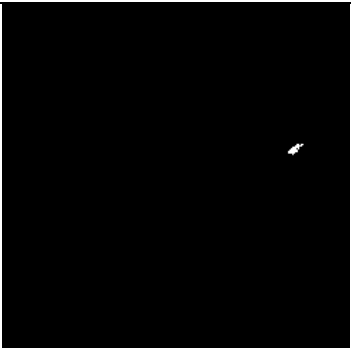
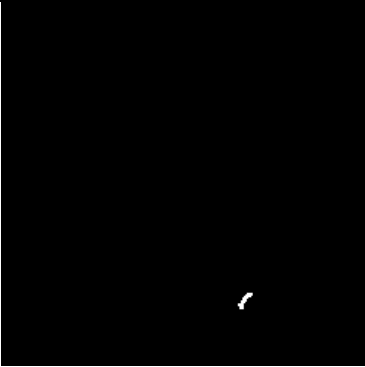
AND SEGMENTATION

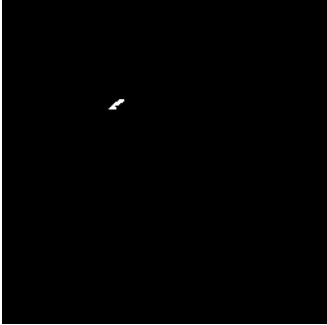
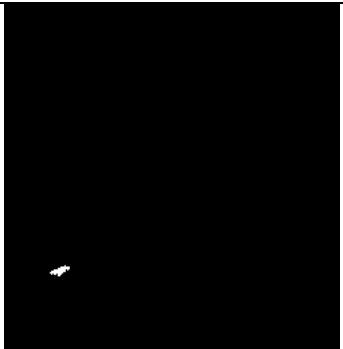
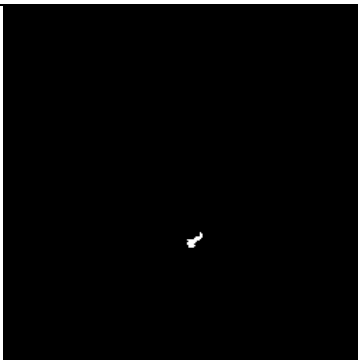
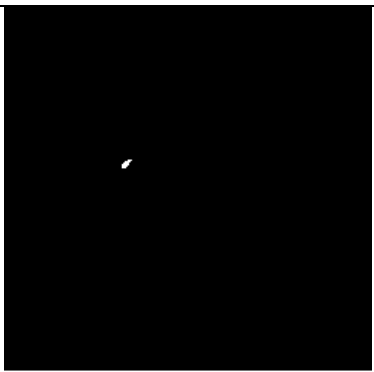
Image Numbers	Input Images	Noise Removed Images
1		
2		
3		

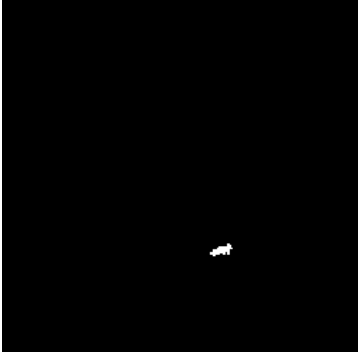
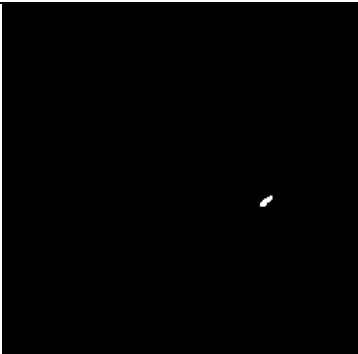
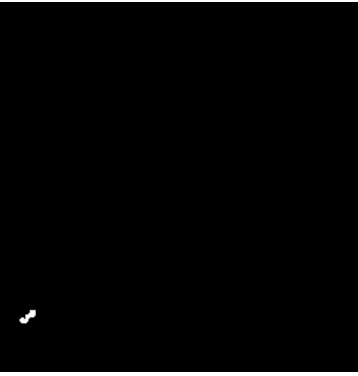
4		
5		
6		
7		

8		
9		
10		

9.2.2 DATASET USED FOR FEATURE EXTRACTION

Image Numbers	SEGMENTED SAR IMAGES
1	
2	
3	

4	
5	
6	
7	

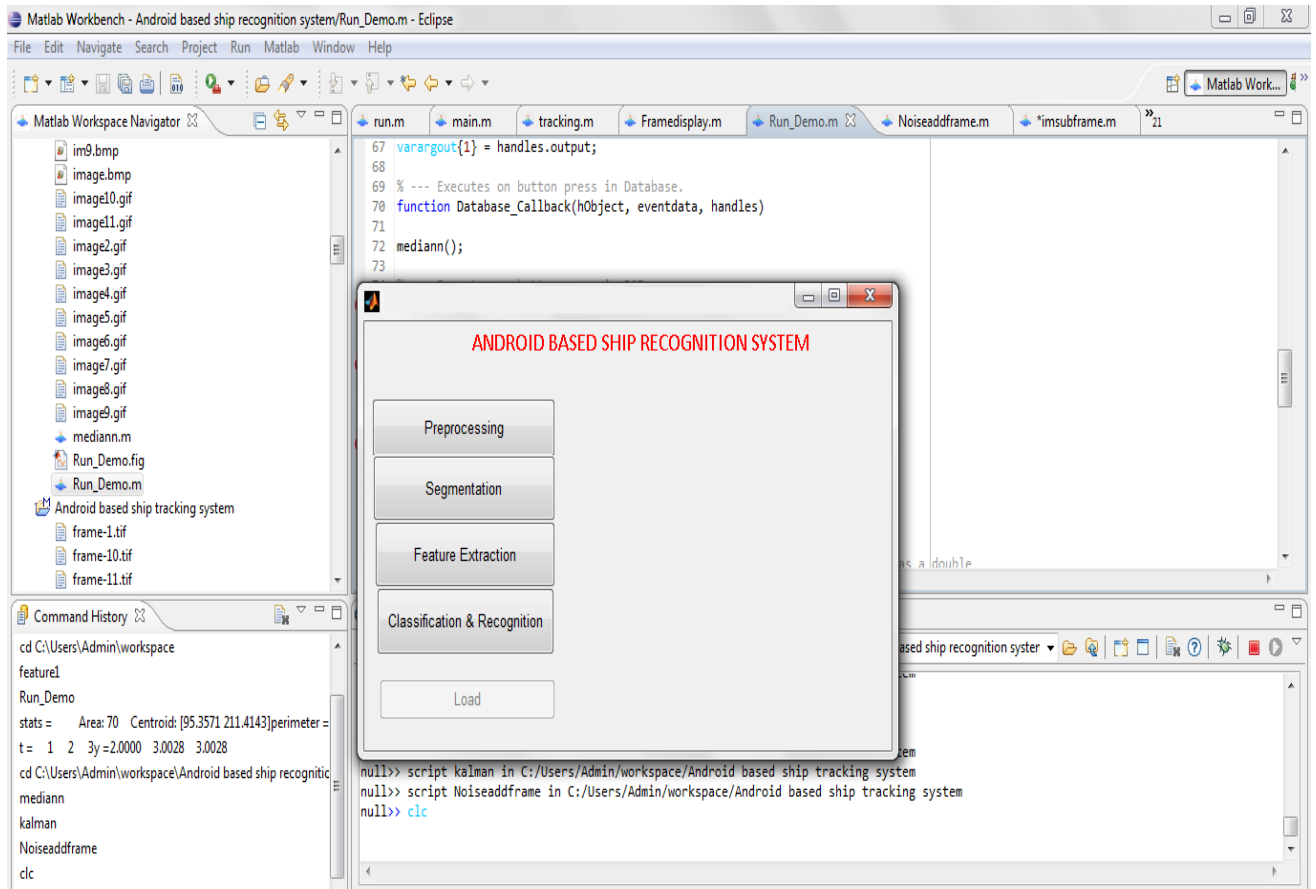
8	
9	
10	

**9.2.3 DATASET USED FOR CLASSIFICATION &
RECOGNITION**

S.No	Area	Perimeter	Centroid x	Centroid y	Mean	Median	Variance
1	70	40.1421	95.3571	211.4143	0.0011	0.0011	0.0334
2	38	24.7279	189.8947	142.7105	5.8671	5.8637	0.0242
3	50	35.7990	134.5800	165.7800	8.0321	8.0258	0.0283
4	39	30.9706	88.6923	81.5641	6.2400	6.2362	0.0250
5	47	32.1421	22.4043	213.1277	7.5200	7.5145	0.0274
6	75	44.3848	154.9200	176.4400	0.0012	0.0012	0.0348
7	41	32.9706	89.6923	83.5641	7.2400	7.2362	0.0250
8	23	17.8995	83.5217	107.5217	3.7247	3.7234	0.0193
9	42	31.5563	164.0952	201.6190	6.9971	6.9923	0.0264
10	38	28.3848	208.4211	104.3158	6.2794	6.2756	0.0251

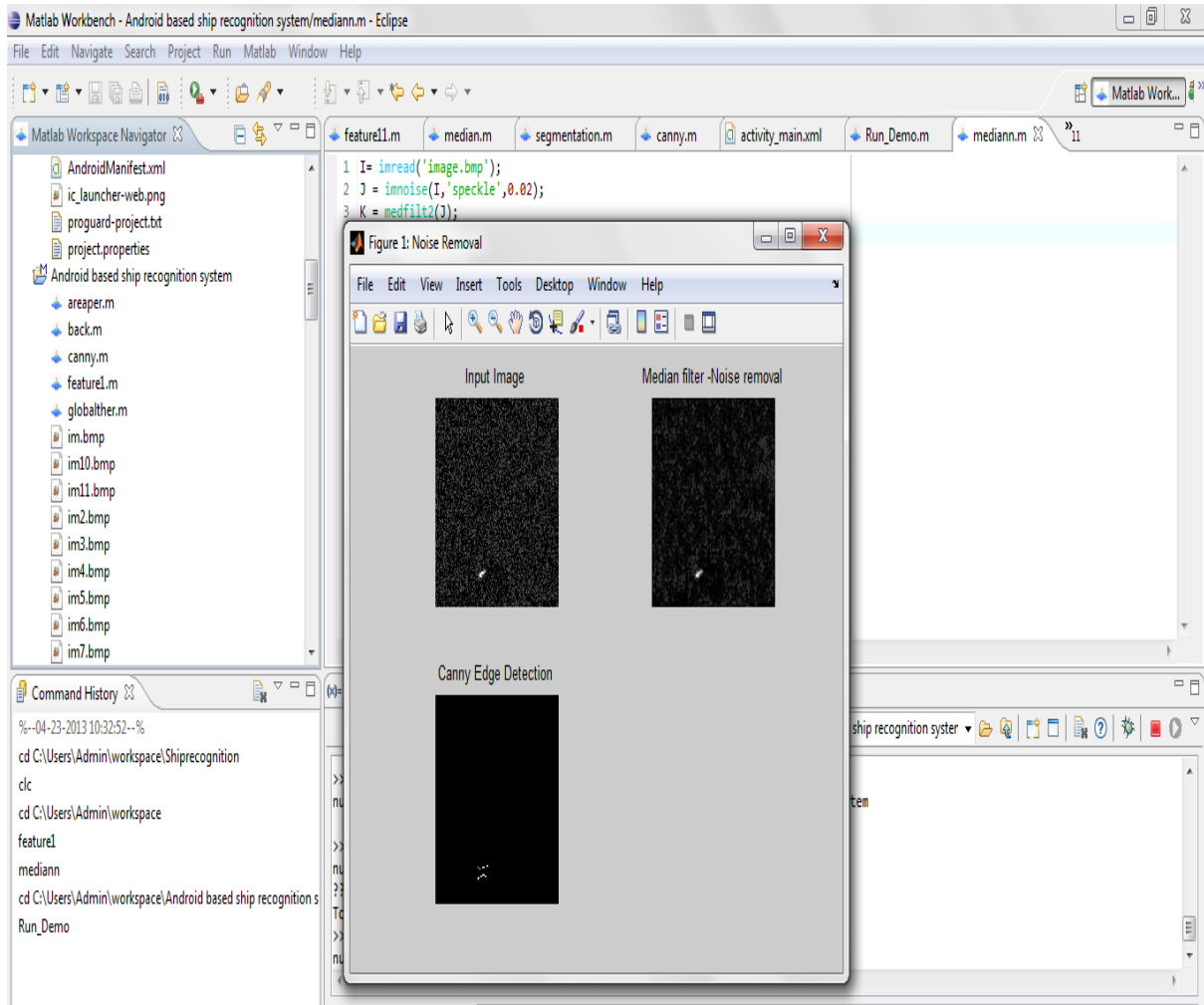
9.3 INPUT SCREENS

9.3.1 MAIN PAGE



9.4 OUTPUT SCREENS

9.4.1 OUTPUT SCREEN FOR PREPROCESSING



9.4.2 OUTPUT SCREEN FOR SEGMENTATION

The screenshot displays the Matlab Workbench environment for an Android-based ship recognition system. The interface includes a workspace navigator, a code editor, a command history window, and a figure window.

Code Editor (segmentation.m):

```
1 f=imread('image.bmp');
2 f=im2double(f);
3 %global Thresholding
4 T=0.5*(min(f(:))+max(f(:)));
5 done=false;
6 while ~done
7     g=f>T;
8     Tn=0.5*(min(g)+max(g));
9     done=abs(T-Tn)<0.001;
10    T=Tn;
11 end
12 display('Thresholding Done')
13 T
14 r=im2bw(f,T);
15 figure,imshow(r);
16 figure,imshow(f);
17 Th=graythresh(f);
18
```

Figure 4: Global Thresholding - Iterative Method

The figure window displays a dark image, likely the result of the segmentation process. The title of the window is "Global Thresholding - Iterative Method".

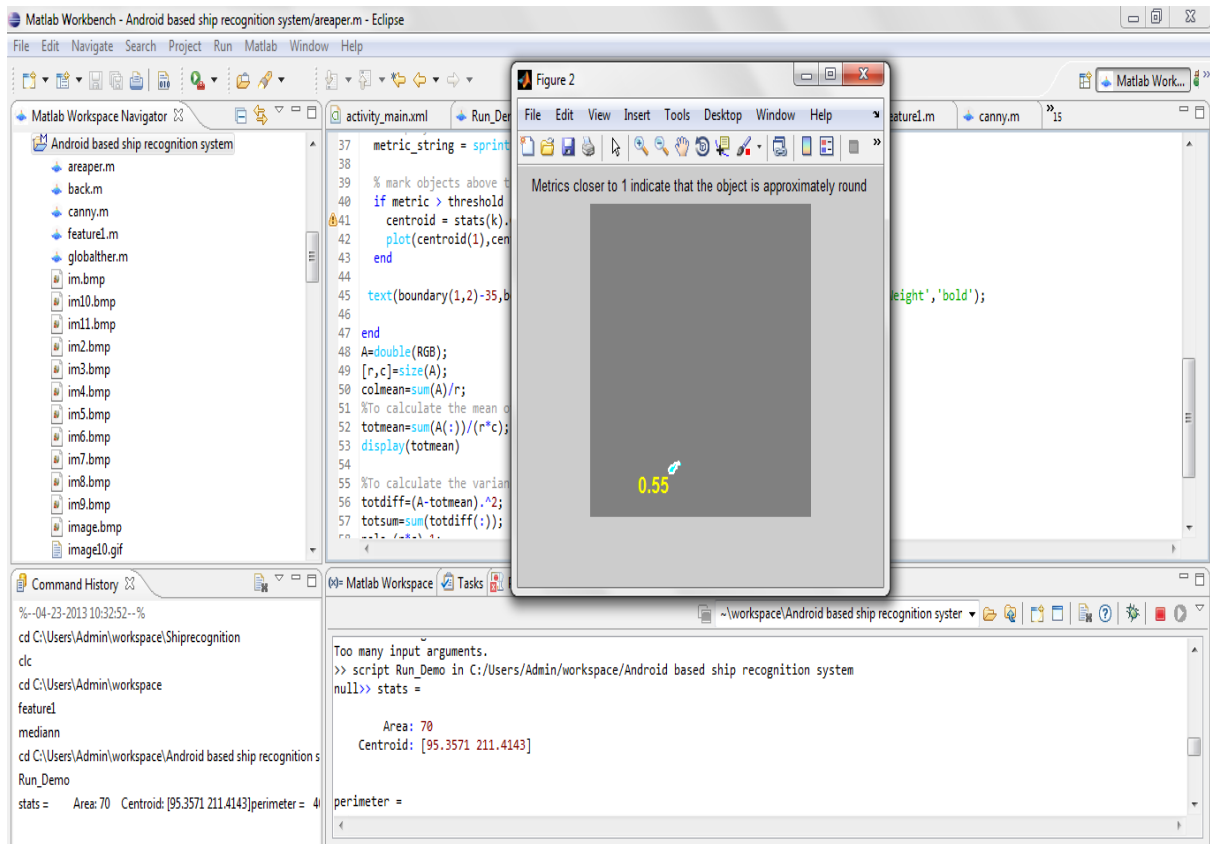
Command History:

```
%--04-23-2013 10:32:52--%
cd C:\Users\Admin\workspace\Shiprecognition
clc
cd C:\Users\Admin\workspace
feature1
mediann
cd C:\Users\Admin\workspace\Android based ship recognition system
Run_Demo
```

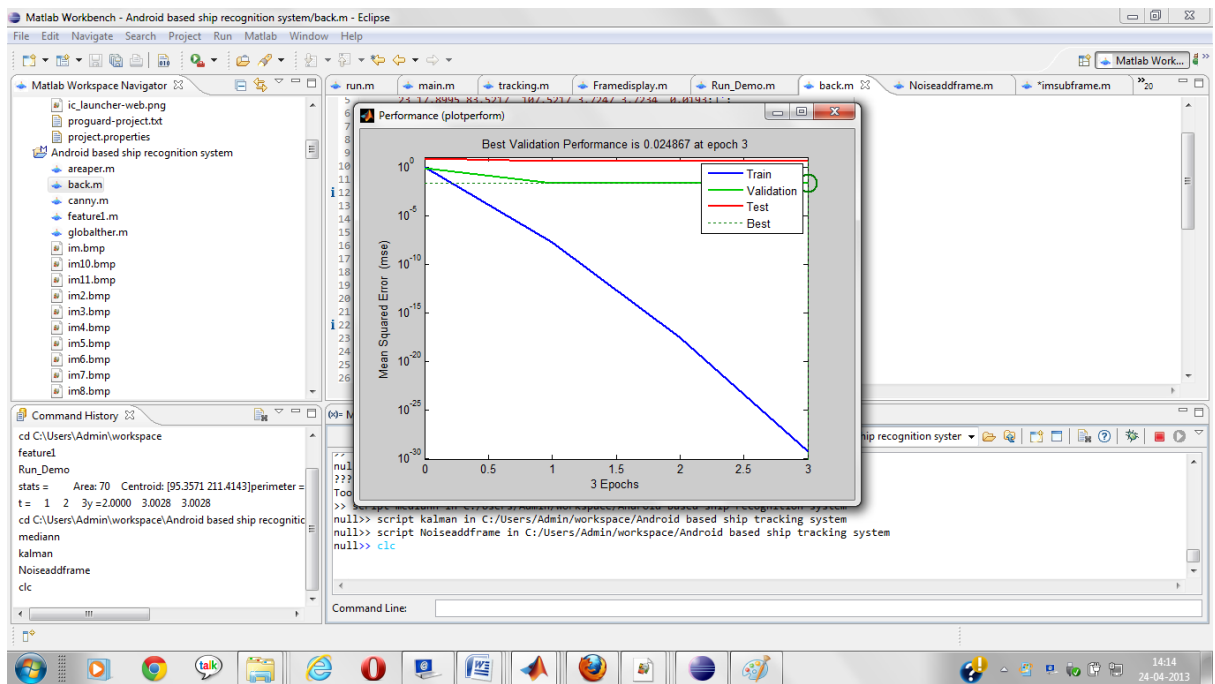
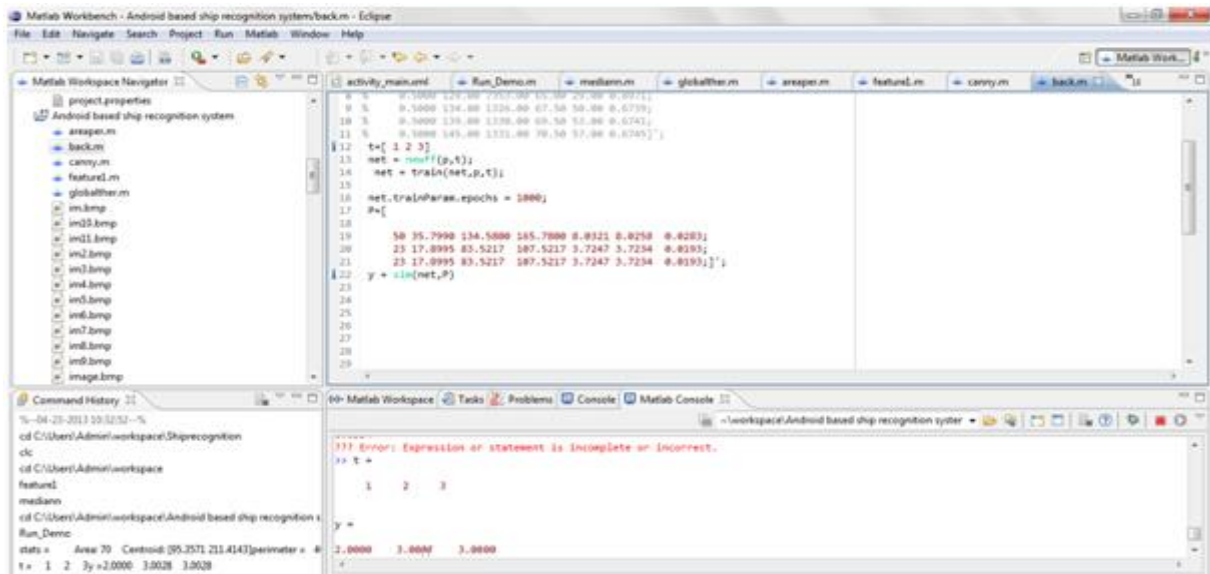
Command Window:

```
>> script Run_Demo
null>> script mediann
null>> script Run_Demo
null>> cd C:\Users\Admin\workspace\Android based ship recognition system
??? Error using
Too many input arguments
>> script Run_Demo in C:\Users\Admin\workspace\Android based ship recognition system
null
```

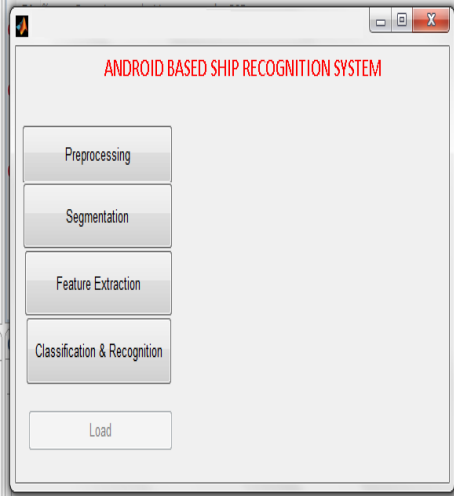
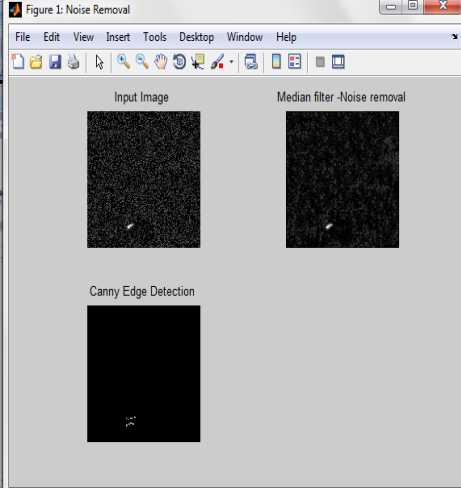
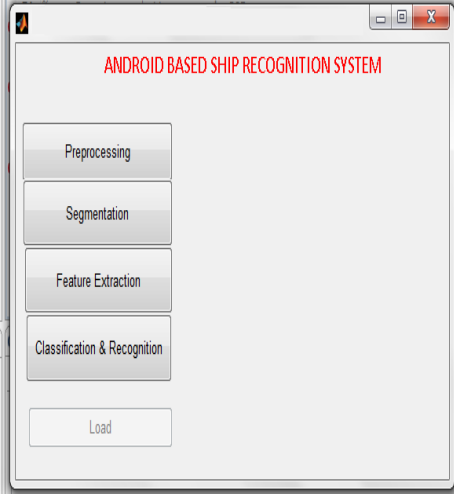
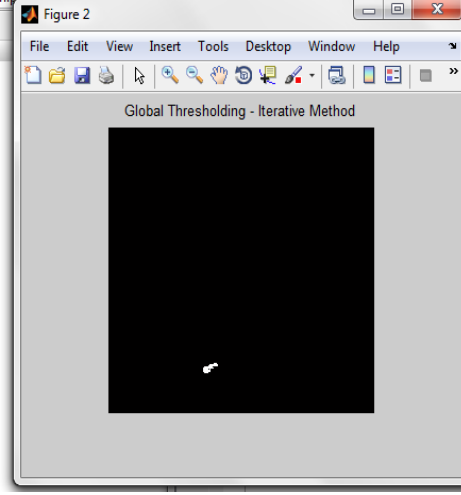
9.4.3 OUTPUT SCREEN FOR FEATURE EXTRACTION



9.4.4 OUTPUT SCREEN FOR SHIP CLASSIFICATION AND RECOGNITION

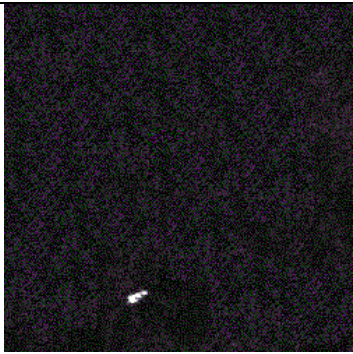
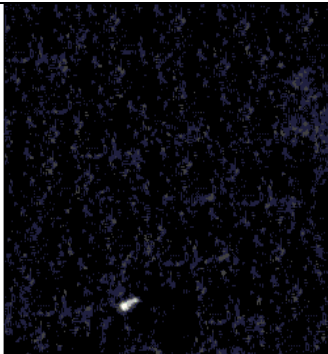
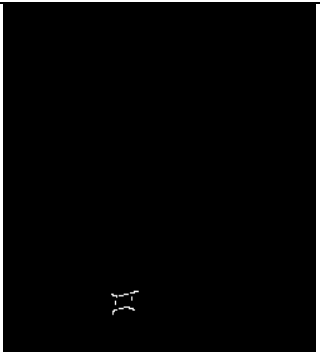
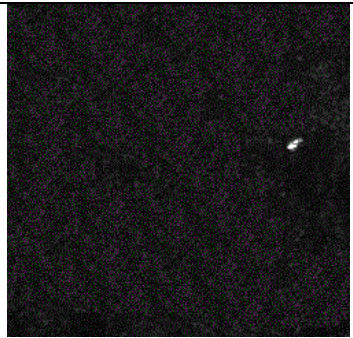
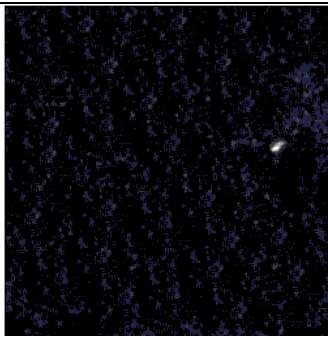
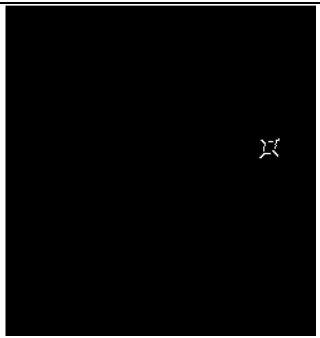
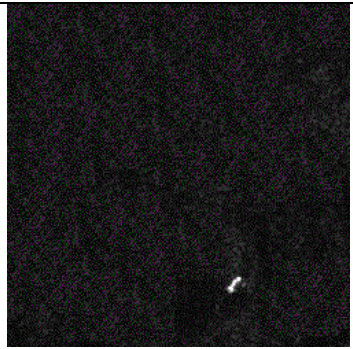
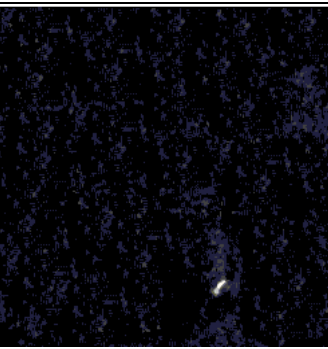
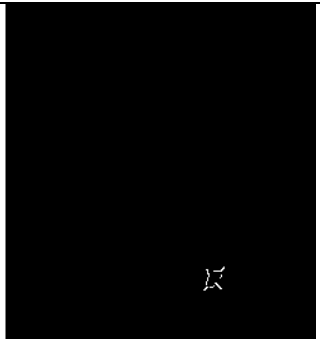

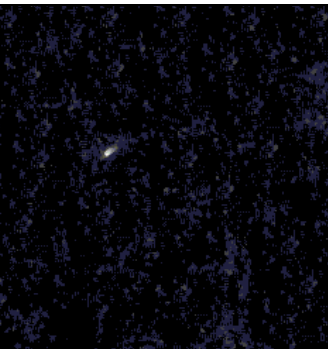
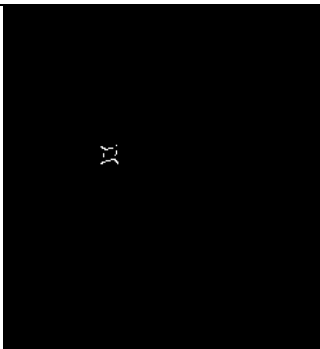



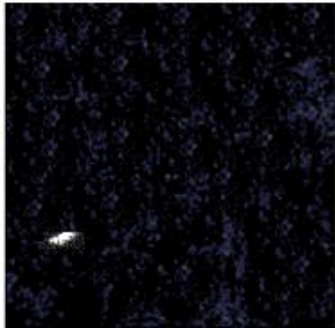

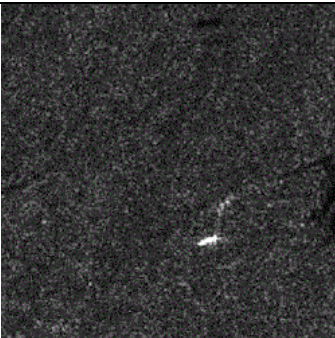
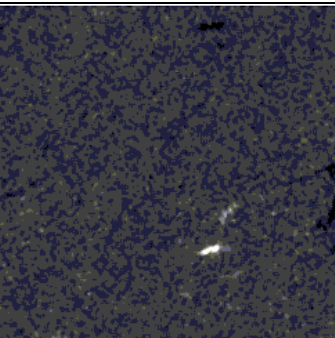


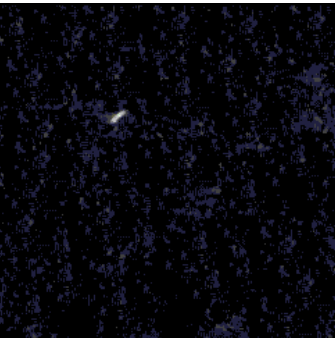
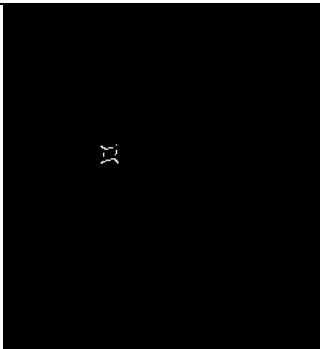
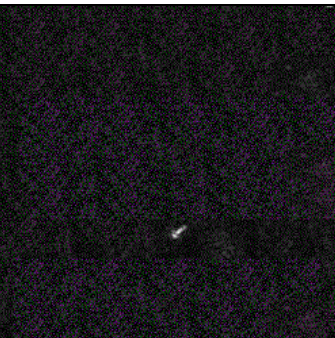
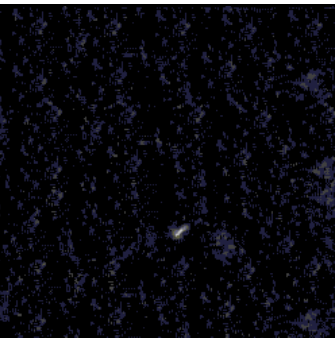
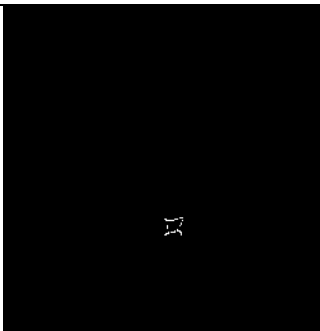
9.5 GUI SNAPSHOTS

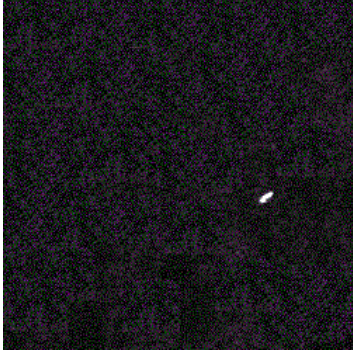
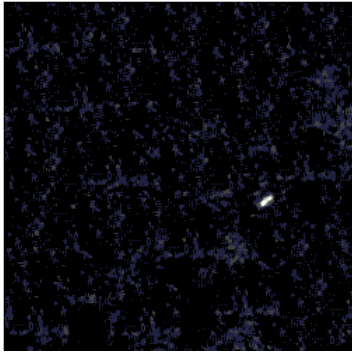
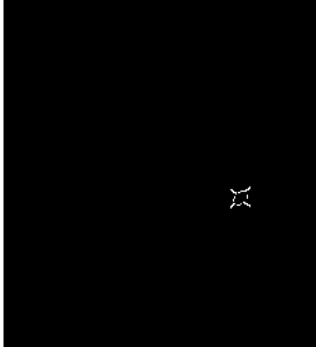
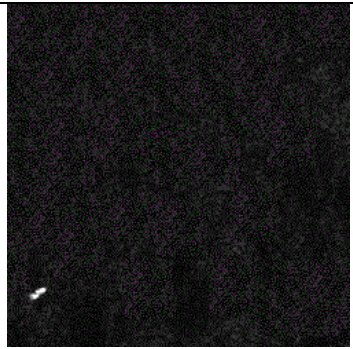
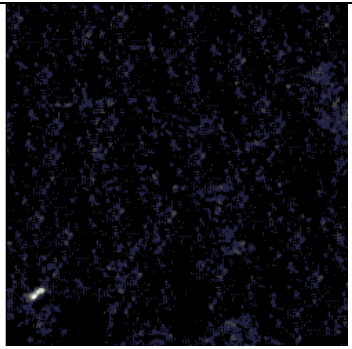
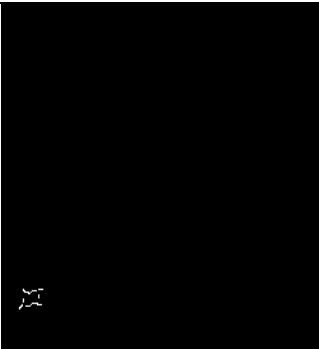
METHODOLOGY	GUI INPUT	GUI OUTPUT
PREPROCESSING	 <p>The screenshot shows a window titled "ANDROID BASED SHIP RECOGNITION SYSTEM" with a red title bar. It contains five buttons: "Preprocessing", "Segmentation", "Feature Extraction", "Classification & Recognition", and "Load". The "Preprocessing" button is highlighted with a grey background, indicating it is the active step.</p>	 <p>The screenshot shows a window titled "Figure 1: Noise Removal" with a standard Windows menu bar (File, Edit, View, Insert, Tools, Desktop, Window, Help). It displays three image processing results: "Input Image" (a noisy grayscale image), "Median filter -Noise removal" (the same image with reduced noise), and "Canny Edge Detection" (the edges of the ship highlighted in white on a black background).</p>
SEGMENTATION	 <p>The screenshot shows the same "ANDROID BASED SHIP RECOGNITION SYSTEM" window. In this step, the "Segmentation" button is highlighted with a grey background, indicating it is the active step.</p>	 <p>The screenshot shows a window titled "Figure 2" with a standard Windows menu bar. It displays the result of "Global Thresholding - Iterative Method", showing a binary image where the ship is represented by white pixels on a black background.</p>

9.6 RESULTS

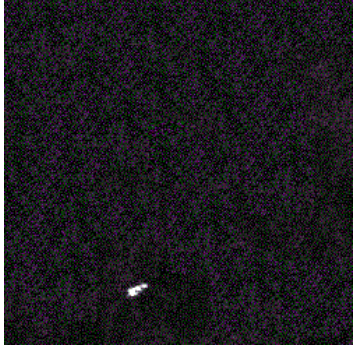
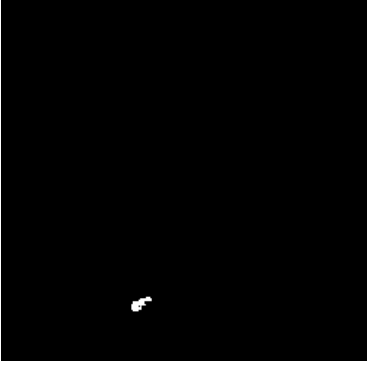
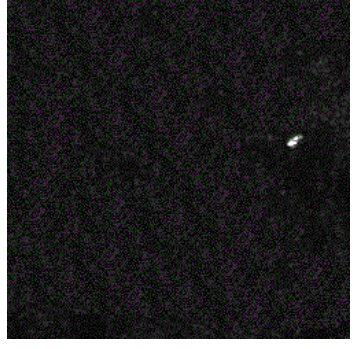
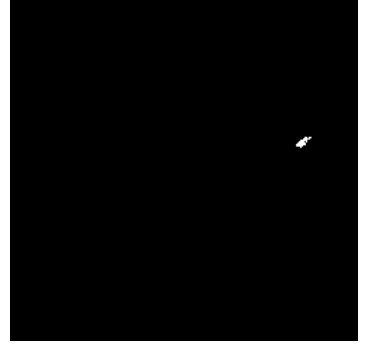
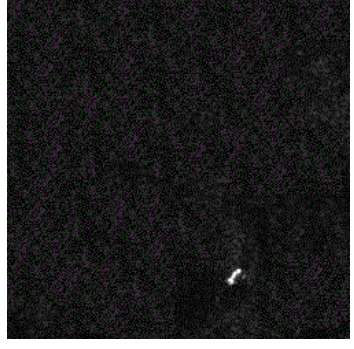
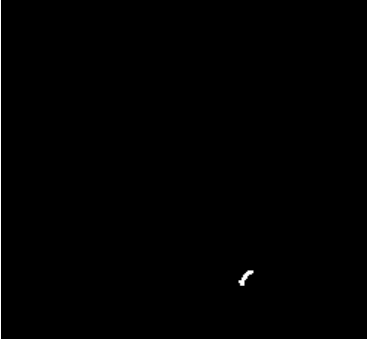
9.6.1 INPUT SAR IMAGES AND PREPROCESSED IMAGES


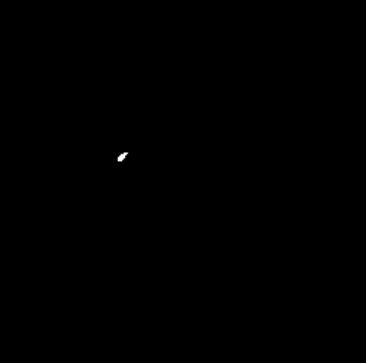


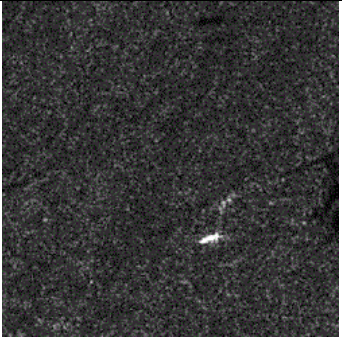
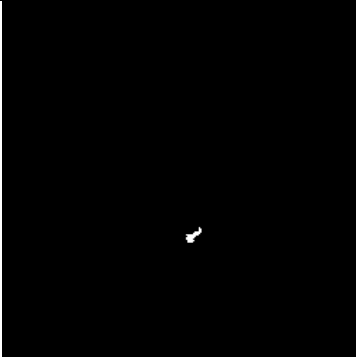

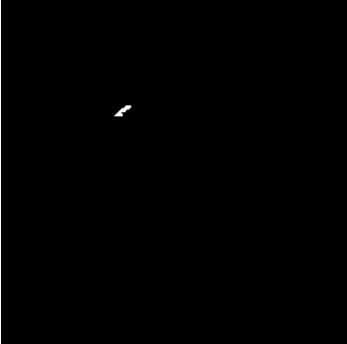
Image Numbers	Input Images	Noise Removed Images	Edge detected Images
1			
2			
3			
4			

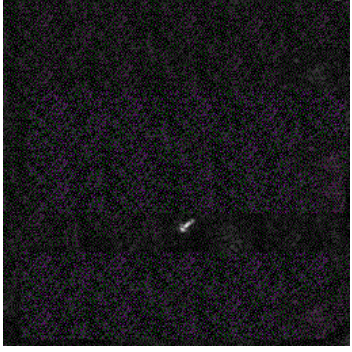
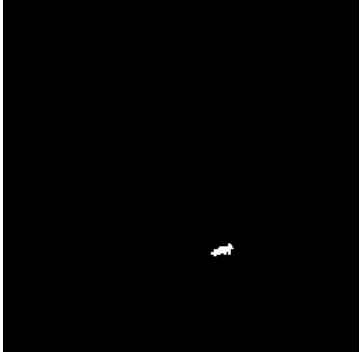
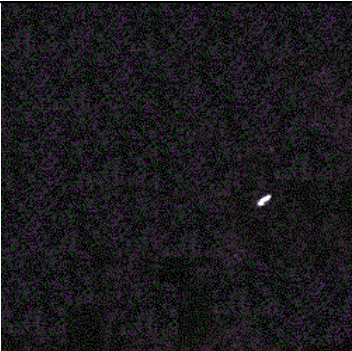
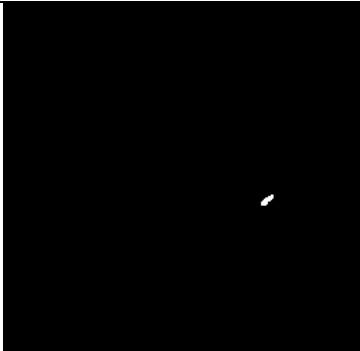
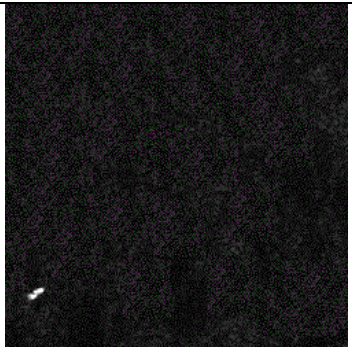
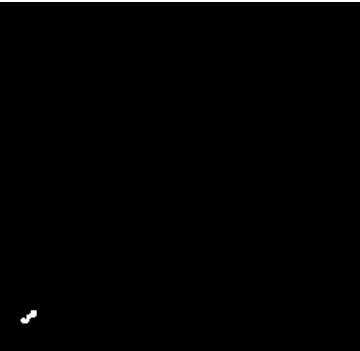
5			
6			
7			
8			

9			
10			

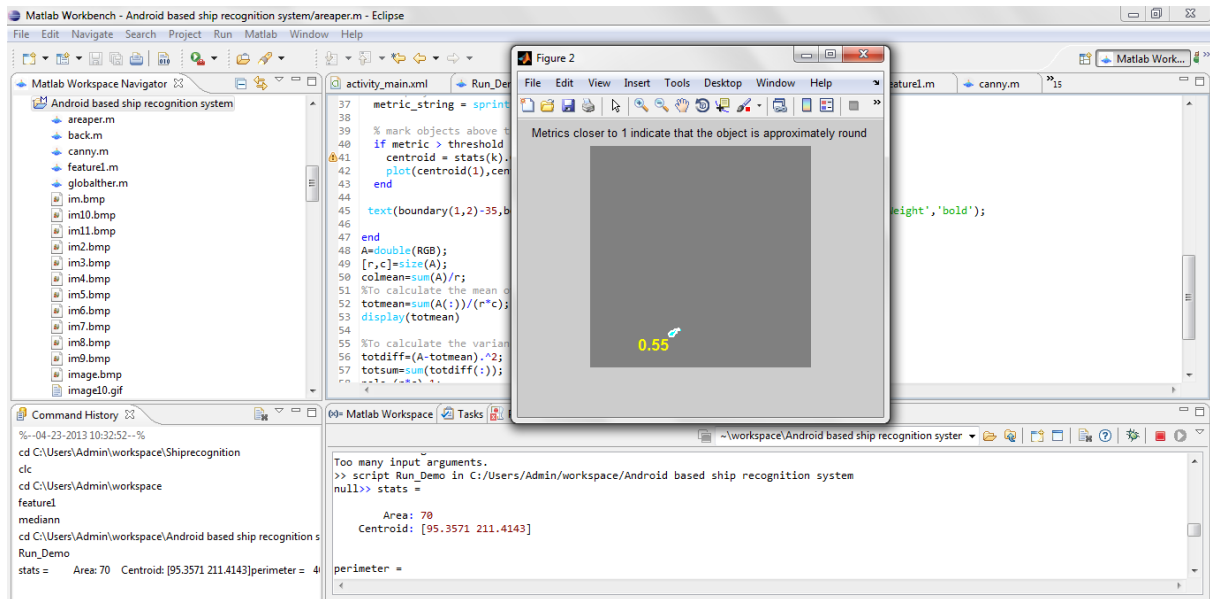
9.6.2 SEGMENTATION

IMAGE NUMBERS	INPUT IMAGES	SEGMENTED SAR IMAGES
1		
2		
3		

4		
5		
6		
7		

8		
9		
10		

9.6.3 FEATURE EXTRACTION



DATASET EXTRACTED FROM THE IMAGE

S.No	Area	Perimeter	Centroid x	Centroid y	Mean	Median	Variance
1	70	40.1421	95.3571	211.4143	0.0011	0.0011	0.0334
2	38	24.7279	189.8947	142.7105	5.8671	5.8637	0.0242
3	50	35.7990	134.5800	165.7800	8.0321	8.0258	0.0283
4	39	30.9706	88.6923	81.5641	6.2400	6.2362	0.0250
5	47	32.1421	22.4043	213.1277	7.5200	7.5145	0.0274

9.6.4 OUTPUT FOR CLASSIFICATION AND RECOGNITION OF SHIPS

The screenshot shows the MATLAB workspace with the following code in the editor:

```

7 % p = [0.5000 122.00 8852.00 61.50 33.50 0.8410;
8 % 0.5000 129.00 7353.00 65.00 29.00 0.8971;
9 % 0.5000 134.00 1326.00 67.50 50.00 0.6739;
10 % 0.5000 139.00 1330.00 69.50 53.00 0.6741;
11 % 0.5000 145.00 1331.00 70.50 57.00 0.6745]';
12 t=[ 1 2 3]
13 net = newff(p,t);
14 net = train(net,p,t);
15
16 net.trainParam.epochs = 1000;
17
18 P=[
19 70 40.1421 95.3571 211.4143 0.0011 0.0011 0.0334;
20 50 35.7990 134.5800 165.7800 8.0321 8.0258 0.0283;
21 23 17.8995 83.5217 107.5217 3.7247 3.7234 0.0193;
22 70 40.1421 95.3571 211.4143 0.0011 0.0011 0.0334;
23 39 30.9706 88.6923 81.5641 6.2400 6.2362 0.0250;
24 47 32.1421 22.4043 213.1277 7.5200 7.5145 0.0274;
25 23 17.8995 83.5217 107.5217 3.7247 3.7234 0.0193;];
26
27 y = sim(net,P)
28
29

```

The MATLAB Console shows the following output:

```

>> t =
     1     2     3

y =
1.0000    2.0000    3.0000    1.0000    2.7954    2.5340    3.0000

```

The screenshot shows the MATLAB workspace with the same code as above. A performance plot window titled "Performance (plotperform)" is overlaid on the workspace. The plot shows the Mean Squared Error (mse) on a logarithmic scale versus the number of epochs (0 to 3). The plot includes four data series: Train (blue solid line), Validation (green solid line), Test (red solid line), and Best (dotted black line). The training error decreases significantly over the three epochs, while the validation and test errors remain relatively constant. The best performance is achieved at epoch 3.

Best Validation Performance is 0.25664 at epoch 3

Epoch	Train (mse)	Validation (mse)	Test (mse)	Best (mse)
0	10 ⁰	~10 ⁰	~10 ⁰	~10 ⁰
1	~10 ⁻⁵	~10 ⁰	~10 ⁰	~10 ⁰
2	~10 ⁻¹⁰	~10 ⁰	~10 ⁰	~10 ⁰
3	~10 ⁻¹⁵	~10 ⁰	~10 ⁰	~10 ⁰