

USC-SIPI REPORT #422

Autonomous Ship Recognition From Color Images

by

Deniz Kumlu

August 2012

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Viterbi School of Engineering
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Suite 400
Los Angeles, CA 90089-2564 U.S.A.

AUTONOMOUS SHIP RECOGNITION FROM COLOR IMAGES

by

Deniz Kumlu

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(ELECTRICAL ENGINEERING)

August 2012

To My Family

Acknowledgements

I hereby want to acknowledge:

- My advisor Prof Keith Jenkins for his insights with the project, ability to solve design and engineering problems and continuous support during my life at USC. He is not only a professor but more than that a role model, a mentor. Thank you for your guidance, patience and hospitality.
- Turkish Navy for financially supporting me during my education. Thank you for making this possible. I am very proud of being a Turkish Naval Officer.
- Jiangyang Zhang , for his friendship, hospitality and ability to look at the project from a broader perspective. He also had continuous support during my masters and made this experience unique for me. Thank you for editing and challenging my thesis.
- I would like to express my gratitude to Bahri Maras, M. Batuhan Gundogdu, Kubilay Savci for making my graduate stay enjoyable.
- My great family, my mother Mujgan, my father Nizam, my brother Kurtulus and my sister Pinar. I am very lucky for having such a loving and caring family. This wouldn't be possible without your continuous support and ability to calm me down. Your guidance enlightened my way.
- And finally my girlfriend Nil. I am sorry for being away from you for two years but I hope this will enforce our relationship and provide us better opportunities in the future. Thank you for your continuous support. You are the reason for me to stick to my goal.

Table of Contents

Dedication	ii
Acknowledgements	iii
List of Figures	vi
List of Tables	viii
Abstract	ix
Chapter 1 Introduction.....	1
1.1 Objective.....	1
1.2 Challenges	2
1.3 Our Approach	3
1.2 Outline of Thesis.....	5
Chapter 2 Construction of Database.....	7
2.1 Synthetic and Real Images	7
Chapter 3 Segmentation	13
3.1 Why Segmentation?	13
3.2 GrabCut Segmentation Algorithm	16
3.3 GrowCut Segmentation Algorithm	17
3.4 Proposed Algorithm	19
3.5 Centering and Resizing	25
Chapter 4 Feature Extraction Model.....	28
4.1 Zoned Based Feature Extraction Model.....	28
4.2 Modified Spatial Pyramid Feature Extraction Model.....	30
Chapter 5 Feature Reduction and Feature Selection.....	34
5.1 Dimensionality Reduction	34
5.2 PCA	35
5.3 Random Projection.....	36
5.4 Sequential Forward Selection	39
Chapter 6 Classifiers and Classification Results	42
6.1 Support Vector Machines	42
6.2 Classification Results for Training Data	46
6.3 Classification Results for Testing Data.....	50
6.4 Comparison of Results with Similar Works.....	54

Chapter 7	Conclusion	55
7.1	Concluding Remarks	55
7.2	Future Work.....	56
References	58

List of Figures

Figure 1 Sample synthetic ship images from previous works [3]	9
Figure 2 Rotating models in the Google SketchUp®	10
Figure 3 Exporting models in the Google SketchUp®	10
Figure 4 Sample synthetic images in our database	11
Figure 5 Sample real images in our database	12
Figure 6 Importance of global details [22].....	13
Figure 7 Different image formats for feature extraction.....	14
Figure 8 Edge detected synthetic ship images.....	15
Figure 9 GrabCut segmentation algorithm results.....	17
Figure 10 GrowCut segmentation algorithm results.....	18
Figure 11 Canny edge detection result.....	20
Figure 12 Distinctive line between sea and ship's superstructure.....	22
Figure 13 Results for cleaning the edge detected image.....	23
Figure 14 Low pass filtering operation for the synthetic image.....	24
Figure 15 Sample edge detected real images in our database	25
Figure 16 Example of different position and different size of ship images	26
Figure 17 Centering and resizing real image in our database	27
Figure 18 Zone based feature extraction method	28
Figure 19 Reducing the feature vector by cropping the zero values	29
Figure 20 Projection in column orientation.....	31
Figure 21 Projection in row orientation	31

Figure 22 Modified spatial pyramid histogram $L = 2$	32
Figure 23 Feature selection	39
Figure 24 Support vector machine for seperable case	42
Figure 25 Support vector machine for non-separable case	44
Figure 26 Non-linear mapping for SVM classifier	45
Figure 27 Feature reduction (Random projection)	47
Figure 28 Feature reduction (SFS)	48

List of Tables

Table 1 Results for training set vs. development set	46
Table 2 Result for random projection	48
Table 3 Result for random projection + feature selection.....	48
Table 4 Result for feature selection	49
Table 5 Result for 10-fold cross-validation.....	49
Table 6 Results for training on synthetic data-set only; testing on test data-set	50
Table 7 Results for training on combined (synthetic and real) data-set; testing on test data-set	51
Table 8 Confusion matrix for training on synthetic data-set only; testing on test data-set	52
Table 9 Confusion matrix for training on combined (synthetic and real) data-set; testing on test data-set.....	53

Abstract

Autonomous ship recognition is an active area for military and commercial applications like harbor surveillance. Accurate identification of unknown contacts is critical in military intelligence. This automated system can help controllers to identify the point of contacts more quickly and accurately. This work mainly focuses on color images attained using digital cameras mounted on ships and harbors. Aside from using digital images for recognition, other information known are distance and course information attained from RADAR. For extracting significant features, spatial pyramid histogram technique is performed on a segmented ship image and support vector machines are used as a classifier. These particular data-sets contain 9 different types of ship with 18 different camera angle perspectives for training set, development set and testing set. The training data-set contains modeled synthetic images; development and testing data-sets contain real images. This work reports two experimental results for ship classification from color images. Our first experiment is based on classification of a synthetic image data-set versus real image data-set, which means the classifier is trained on the synthetic data-set and tested on the real data-set and obtained accuracy is 87.8%. Our second experiment is based on classification of synthetic images + real images (combined data-set) versus real images, which means the classifier is trained on the combined data-set and tested on a separate real data-set, and obtained accuracy is 93.3%.

Chapter 1

INTRODUCTION

1.1 Objective

Autonomous ship recognition can provide quick and accurate information to the controllers for identifying the ship's class. Such information is important for military applications and is also useful for harbors interested in implementing better tracking and automatic monitoring. Most ships are equipped with high quality dome cameras and fixed cameras for observation purposes around the bridge of the ship. The watch officer can control these cameras with a joystick having access to a full 360° angle view. Similar cameras are used in harbors having the same monitoring system for surveillance. Current monitoring systems for identifying ships are human based. Trained officers compare silhouettes of ships seen from cameras, both on harbors and bridges, against images stored in Jane's Fighting Ships database reference book. Obvious drawbacks of this current system for matching ships are speed of recognition, human error, and lack of experienced officers.

This thesis proposes a system for object recognition of ships using algorithms for edge and line detection, noise removal, feature extraction and multiclass object classification. Using classification methods will help watch officers and harbor controllers identify the ship types and hopefully provide a step towards building an automated ship identification system. This work can help officers and harbor controllers

narrow the search area for detected ships, rather than browsing through the entire database of known ships.

1.2 Challenges

The first challenging part of this work is constructing a current ship image database gathered from images on the internet. Because currently there is no available database for ship images, we constructed an image database. Images from the internet tend to not have enough warship images available to the public with different angle views in azimuth. Another problem is that most of the ship pictures available are from front view or side view. This limits the training dataset to a limited amount of camera angles to process. However, in the real case, ships are observed from any angle. In this work, the data-sets for real images are limited to $\pm 40^\circ$ from side views, no frontal views are taken into account. Since the front view of the ships generally do not have much information for identification, it is nearly impossible to identify ships solely from a frontal view. For accurate system results, the data-set must contain similar amounts of images in every azimuth angle. For example, in our data-set we used 10° resolution, and in this case our data-set should contain at least 2 images or more for every 10° course angle. But, it is difficult to find these images on the internet. Otherwise, if the data-set contains 100 images and just 10 of them are frontal views; this wouldn't reflect the real situation. Therefore, building a strong data-set is important for future work; for now, only side views are used up to a certain azimuth angle.

The second challenging part of this work is segmentation. Simple thresholding fails to segment the ship images. We also tried more advanced algorithms like Grab cut[26], Graph cut[8] and Grow cut[29] to segment the ship images. All of these

advanced methods need user interaction to segment the ship images failing to be completely autonomous. Also, these segmentation methods are not accurate enough for feature extraction. Since warships are highly camouflaged with the sea and the sky, methods like Grab Cut fail to segment them.

The third challenging part is to have a robust feature extraction after segmentation. Working on edge detected images poses a challenge for extracting significant features for accurate identification. There are not many robust methods for feature extraction on complex binary images. Finding significant features for accurate identification is another challenging part of this work.

The last problem is to reduce the feature space without losing accuracy in order to increase the processing speed of recognition. In this work, a large size image is used for edge detection to get detailed superstructure information. Main distinctive information on the ship images are kept in superstructure data. Every ship class has a different superstructure type. Using smaller images when constructing the database can cause loss of important details. On the other hand, using large images result in can high dimensional feature vectors which make our system slower. The trade-off here is that larger images take longer the process while smaller images lead to poor accuracy. Therefore, in order to increase the processing speed of data-sets containing larger images, we employ feature reduction[13] and feature selection[19] methods. These methods also improve the accuracy of our overall system.

1.3 Our Approach

In this work, the aim of the proposed method for segmentation is to make this system autonomous for ship recognition and the reason for combining synthetic 3D

models and real ship image datasets is to solve the limited database problem for warship images with different orientation angles. To get around this limitation, synthetic 2D images were generated using the Google 3D Warehouse models of ships. Instead of creating our own ship models in MATLAB® which is not an optimal tool for modeling, we moved to more professional tools to create synthetic ship images. These synthetic images were produced by using Google SketchUp@[21] and the plug in Rhino® for rendering. As a result, 2D ship images were exported with different orientations in azimuth and elevation.

In order to make our system autonomous, advanced segmentation algorithms were abandoned and edge detected images were used instead. During the edge detection part, we got some help from the ship geometry. First, Sobel filter was used for edge detection; after that connected component analysis was applied for detecting the connected components in the image. Then, Hough transform[12] was conducted to detect the lines in the image. Lastly, simple isolated noise removal techniques were performed to kill outliers.

For feature extraction from edge detected images, a popular approach uses histograms of edges[25] of the underlying images and is known as the histogram-of-gradient approach. These models and their hierarchical versions have been shown to perform well in edge detected images. However, it is hard to apply these methods for binary images. Therefore, some modification has been done on this approach to apply to our case. This modification can be explained as projection of edges in the vertical dimension to get the superstructure information. In this method, modified spatial pyramid[20] was used to get edge information in a hierarchical way.

For the feature reduction and feature selection part, first the principal component analysis (PCA)[17] method was used which performs linear mapping of the data to a lower dimensional space by calculating the variance of the data. But, in this application, well known PCA feature reduction method performed poorly. Hence, in order to get better results, random projection method was applied which maps the high dimensional data-set to lower dimensional space, while providing some guarantees on the approximate preservation of distance. In addition, random projection is computationally very simple compare to PCA and produces good results for sparse matrices. After reducing the dimension with random projection[10], forward feature selection[19] was applied to choose best features out of the transformed feature space. This feature selection method was also applied to our overall feature space to reduce the dimension.

1.4 Outline of Thesis

This thesis is structured as follows. In Chapter 2, we describe how this database was constructed using synthetic and real images and how synthetic images were produced with different resolution of azimuth and elevation angle using Google SketchUp®. Thereafter, the contribution of this work for synthetic images will be compared with similar published works on ship recognition. In Chapter 3, Segmentation methods will be discussed and a brief description will be given about how contemporary methods fail to segment the ship images. We will also describe how to binarize the real images to obtain useful features using connected component analysis[5], Hough Transform[12], and isolated noise reduction methods. In Chapter 4, we will describe two methods for feature extraction: zone based and modified spatial pyramid methods. In Chapter 5, feature reduction and feature selection methods will be described. We will start with very well

known PCA method and thereafter explain and perform random projection and sequential forward selection. In Chapter 6, classification methods will be described to evaluate our feature extraction methods and the SVM[16] classifier will be discussed. The experimental results will be given based on these data-sets, and we will discuss our results and present various key evaluations. Then, our results will be compared with similar published works in the literature. In Chapter 7, we will conclude our work and present some closing remarks along with some of the directions in which this work can be extended.

Chapter 2

Construction of Database

2.1 Synthetic and Real Images

One of our contributions is construction of databases from real and synthetic images. When literature search is made about ship data-sets, there were no available data-sets found for public usage. There are several theses from Naval Post Graduate School (NPGS) which is a school for military purposes, about ship recognition from IR silhouette[3,4,15] and in these theses, they created their own synthetic data-set using MATLAB®. They tried to create 3D images as close as to real images by similarity of silhouette and exported these 3D images into 2D images with different azimuth and elevation resolutions by using this program.

However, to model ship images in MATLAB® is not the optimal way for 3D images. There are 3D CAD® programs which make this job more professional and these programs can model the ships according to their real dimensions. To increase accuracy for classification part, modeled ship should capture every detail for superstructure like in the real version; most distinctive features of the ships are found in superstructure. If the ships can be modeled in a realistic way, these modeled images can help to classify the real ship which is the starting point of our work. For this purpose, when synthetic image database is constructed, Google 3D warehouse is used which makes 3D models available for public usage.

This database contains training and testing data-sets. Training data-set is constructed by synthetic images and real images. The real images in the training set are

called development or tuning set. The reason for using development set is cross validation can't be used to evaluate this training data which is solely constructed by synthetic images. In other words, using cross validation for training set which contains synthetic images will produce perfect results. Because, there are no noise or complex background in synthetic images data-set and it is easy to segment and extract features from this data-set. So, to set up parameters by using cross-validation on synthetic training data will produce misleading results. Therefore, one more data-set is created as a development set and feature extraction, feature reduction and classification are performed on this data-set. This development set is used to get more accurate results in the training part and at the end final parameters are used on the testing data. If only synthetic images set was used for training part, the possible scenario will be like this; high accuracy for training part using cross-validation and poor accuracy for testing part.

For testing data-set, only real images are used. There are totally 9 ship types in this training and testing data-sets. These ship types are cruise vessel, fishing vessel, container ship, oil tanker, sailing vessel, cruiser, battleship cruiser, corvette and frigate. In training data-set, there are 36 images per ship types and 10° degree resolution is used for azimuth angle. These data-sets contain following azimuth angle: 0° , 10° , 20° , 30° , 40° , 140° , 150° , 160° , 170° , 180° , 190° , 200° , 210° , 220° , 320° , 330° , 340° , 350° and two elevation angles 0° , 5° degrees. There are 12 images per ship types in development data-set and 10 images per ship types in testing data-set with different azimuth and elevation angles within the boundaries of above angles.

Fig. 1 shows our contribution to synthetic data. In the previous thesis[3], they created synthetic images like in the Fig. 1. As seen in the figure, these modeled images

are not reflecting the real images even for the silhouette. These models are very poor models for ship classification problem.

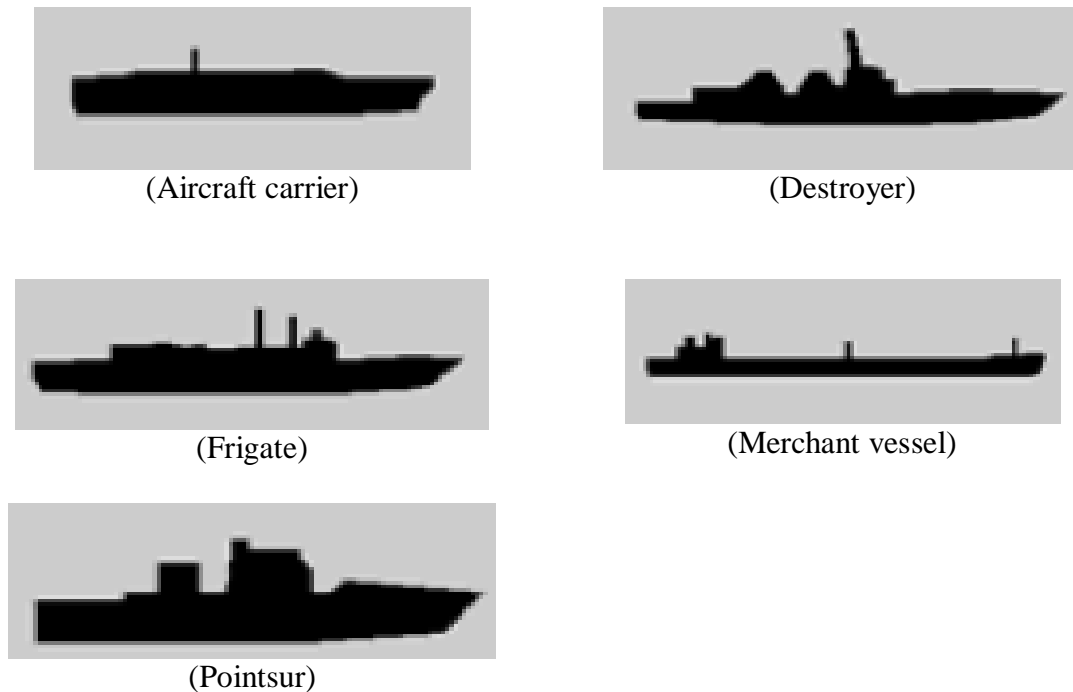


Figure 1: Sample synthetic ship images from previous works [3]

Our synthetic images are created by using Google SketchUp® program. This program is publicly available and can be used for exporting 3D image into 2D image with different azimuth and elevation angles. But, this is not an autonomous program for 2D conversion. For small data-sets and coarse resolution, Google SketchUp® program is reasonable. However, for large data-set and fine resolution, it is not. Because the image is rotating by manually for each angle which is time consuming for fine resolutions. If you want to create images with fine resolutions like 1° in azimuth and elevation, then more sophisticated programs must be used like 3D CAD programs with some plug-in for Google SketchUp®. In this way, Google 3D images can be exported as 2D images in an autonomous way and it helps user to save considerable amount of time for this part.

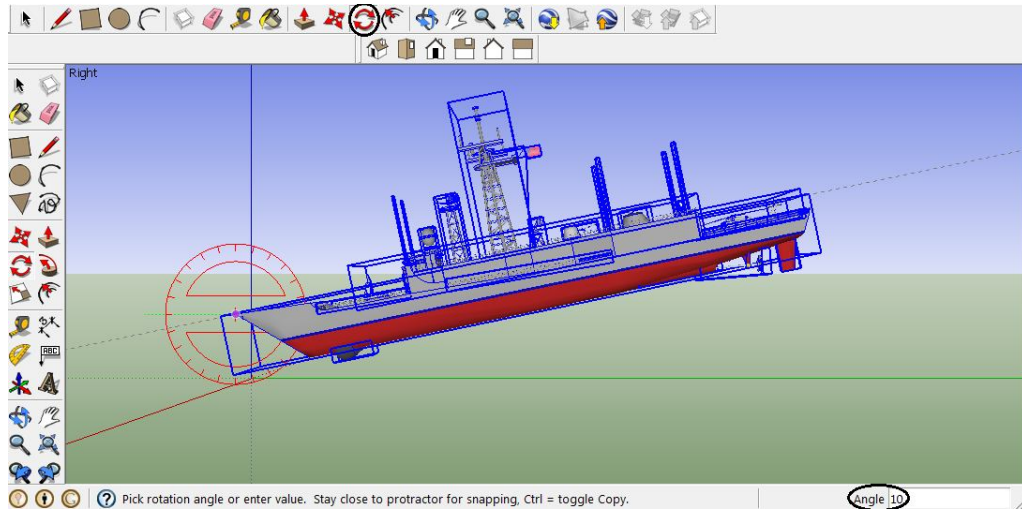


Figure 2: Rotating model in the Google SketchUp®

Fig. 2 is illustrating how the 3D ship model is rotated by using Google SketchUp® program. Simply, there is a rotation button in the upper toolbar which is circled. When this button is clicked, the rotation symbol appears on the screen. The color of the symbol describes the axis that you want to rotate. If the color of the symbol is red as seen in the figure, the model rotates around the red axis. The angle value which is circled can be entered in the lower corner as a number from the keypad. If 10 is entered as an input, the model rotates 10° around the red axis.

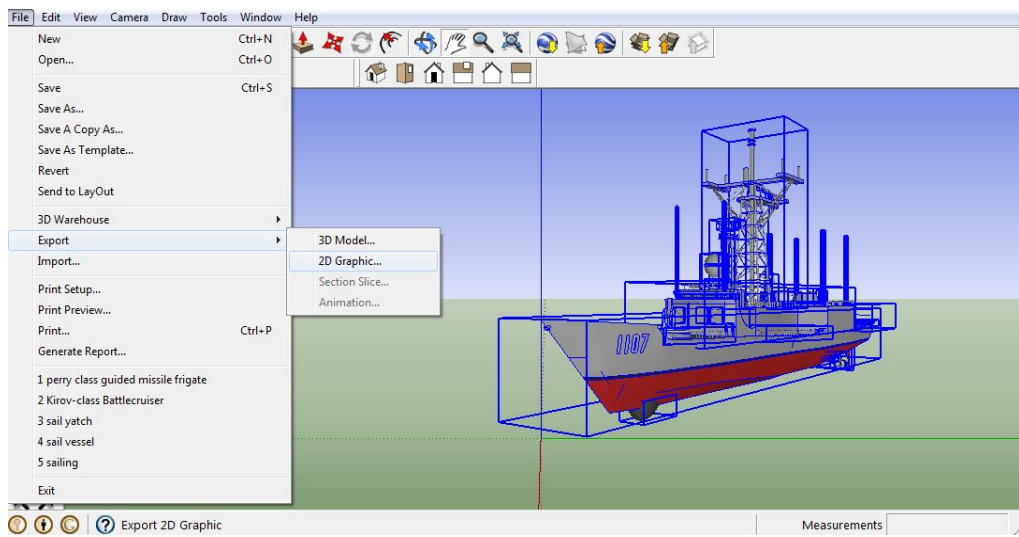


Figure 3: Exporting model in the Google SketchUp®

Fig.3 shows how the 3D ship model can be exported as 2D image after rotation. As we discussed before, for every single course, the model should be rotated and exported by using this procedure which is time consuming for large data-sets or fine resolutions.

Models in Fig. 4 are selected from Google 3D Warehouse which contains variety range of ship types even contains most of the warships, and they reflect the real images as dimensions of height, length and width. In addition, these models have every detail for superstructure in terms of antennas, funnel, bulls eye or canon.

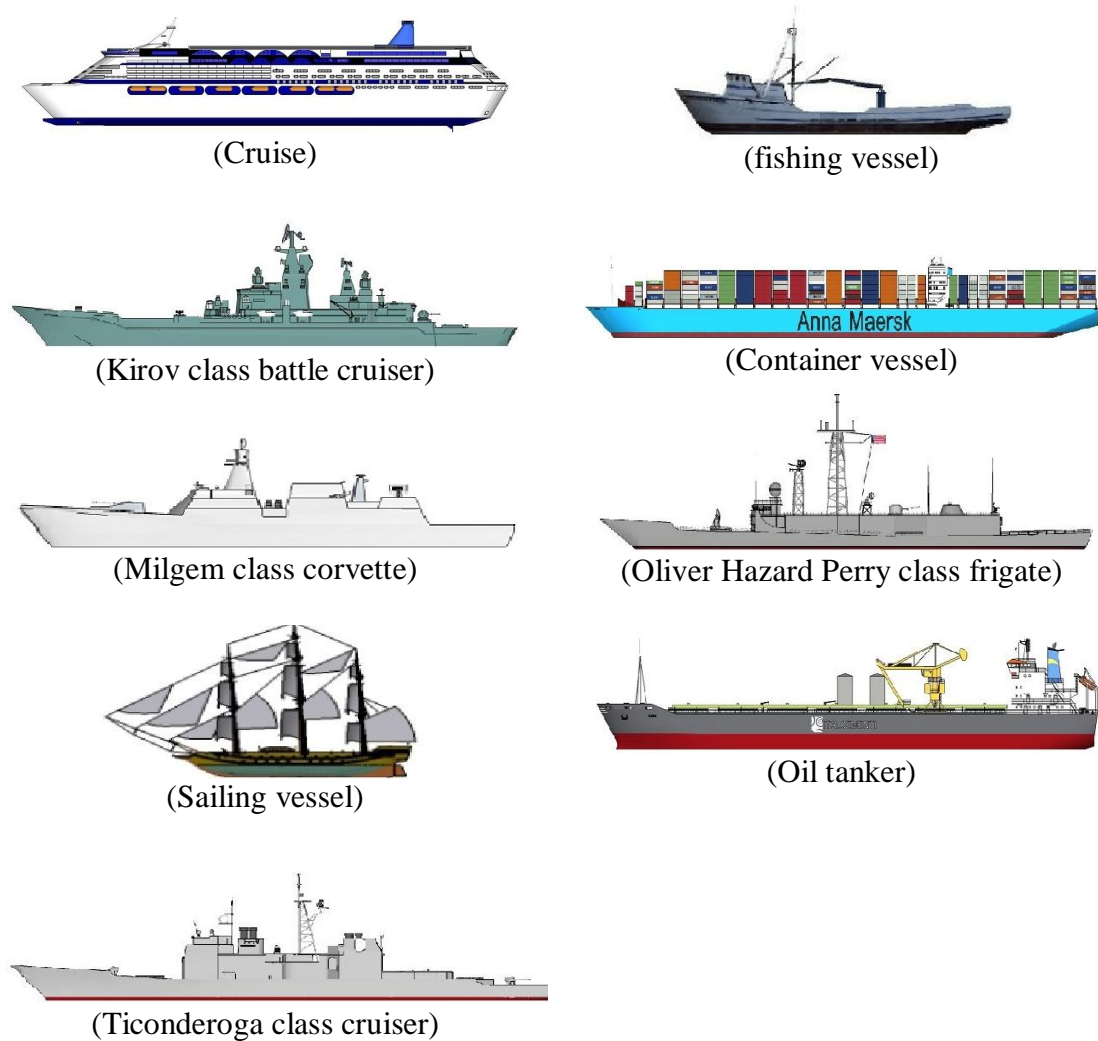


Figure 4: Sample synthetic images in our database

The real images in Fig. 5 are collected from Google image's database. There are not many good quality warship images from different perspectives in this database for intelligence reasons; therefore, 12 images per ship type for development set and 10 images per ship type for testing set were chosen because of this limitation. The images were collected before the segmentation algorithm had been determined; otherwise datasets could be biased. After writing the code for segmentation, it is easy to determine what kinds of images are easy to segment for the algorithm. So, writing the algorithm after data collection can produce more realistic results for this system.



(Cruise)



(Fishing vessel)



(Kirov class battle cruiser)



(Container vessel)



(Milgem class corvette)



(Oliver Hazard Perry class frigate)



(Sailing vessel)



(Oil tanker)



(Ticonderoga class cruiser)

Figure 5: Sample real images in our database

Chapter 3

Segmentation

3.1 Why Segmentation?

For this classification system, several different methods are used for segmentation problem and also some methods are tried like Gist Features[22] and Spatial Pyramids[20] which allow classifying the color images without segmenting them. In terms of what others have reported, it is possible to get 84% accuracy in a scene classification problem, by performing Gist Features[23]. Using these methods for feature extraction can help to skip segmentation which took most of the time for this system; how these methods work for feature extraction can be explained in the following statements. When the image is segmented and binarized via edge detection, most of the information might be lost in the image in terms of feature extraction. Sometimes background of the image can carry some important information.



Figure 6: Importance of global details [22]

Fig. 6 is illustrating the importance of the background for scene categorization. When the left figure is checked, this image can be categorized visually as a street category. Since, our mind classifies the furniture as building because of the street scene context in the left figure. However, when the high resolution image is checked, the furniture can be easily recognized. Here, global features of the image help to classify this image as a street category. These approaches didn't produce good results for ship classification problem for the following reason. The backgrounds of the ship images more or less carry the same type of information like sea and sky. So, using global features could not help classification problem in ship database. As a result, poor classification results were obtained from these methods and we turn back to segmentation problem again which was the most challenging part for this system.

During the segmentation part, we realized that most of the important information exists in the edges; therefore edge detection was preferred instead of using silhouette of the ship. The edge detected images carry much more information than silhouette of the same image in terms of feature extraction as seen in Fig. 7.

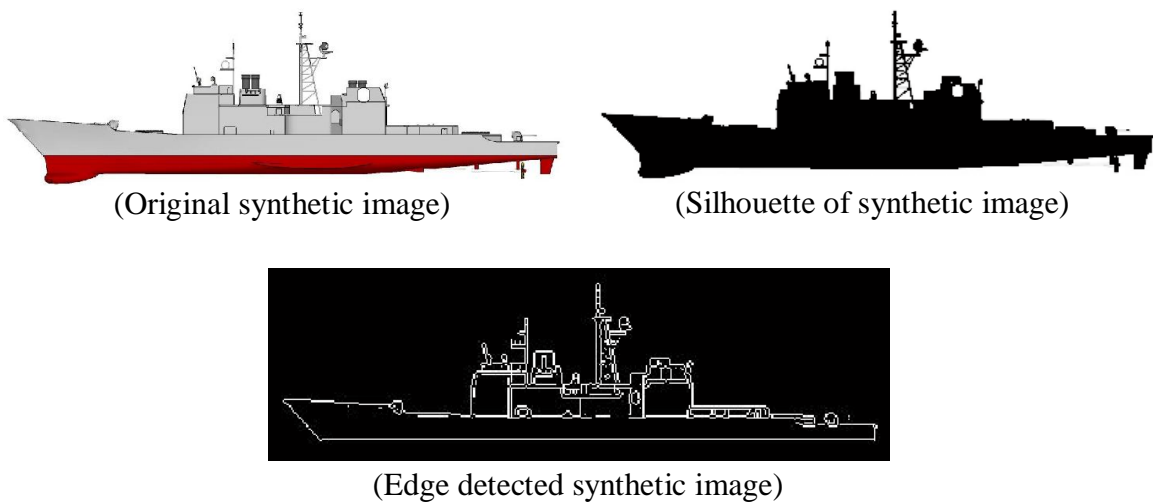


Figure 7: Different image formats for feature extraction

To binarize this data-set is easy for the synthetic images, which don't need any segmentation because of their uniform background and lack of noise. Therefore, using only Sobel edge detector can give a good result for edge detection part. However, in real image data-set, additional processing is necessary to clean the image after using the Sobel edge detector for noise and complex background issues. The edge detected results of the different oriented synthetic images are shown in Fig. 8.

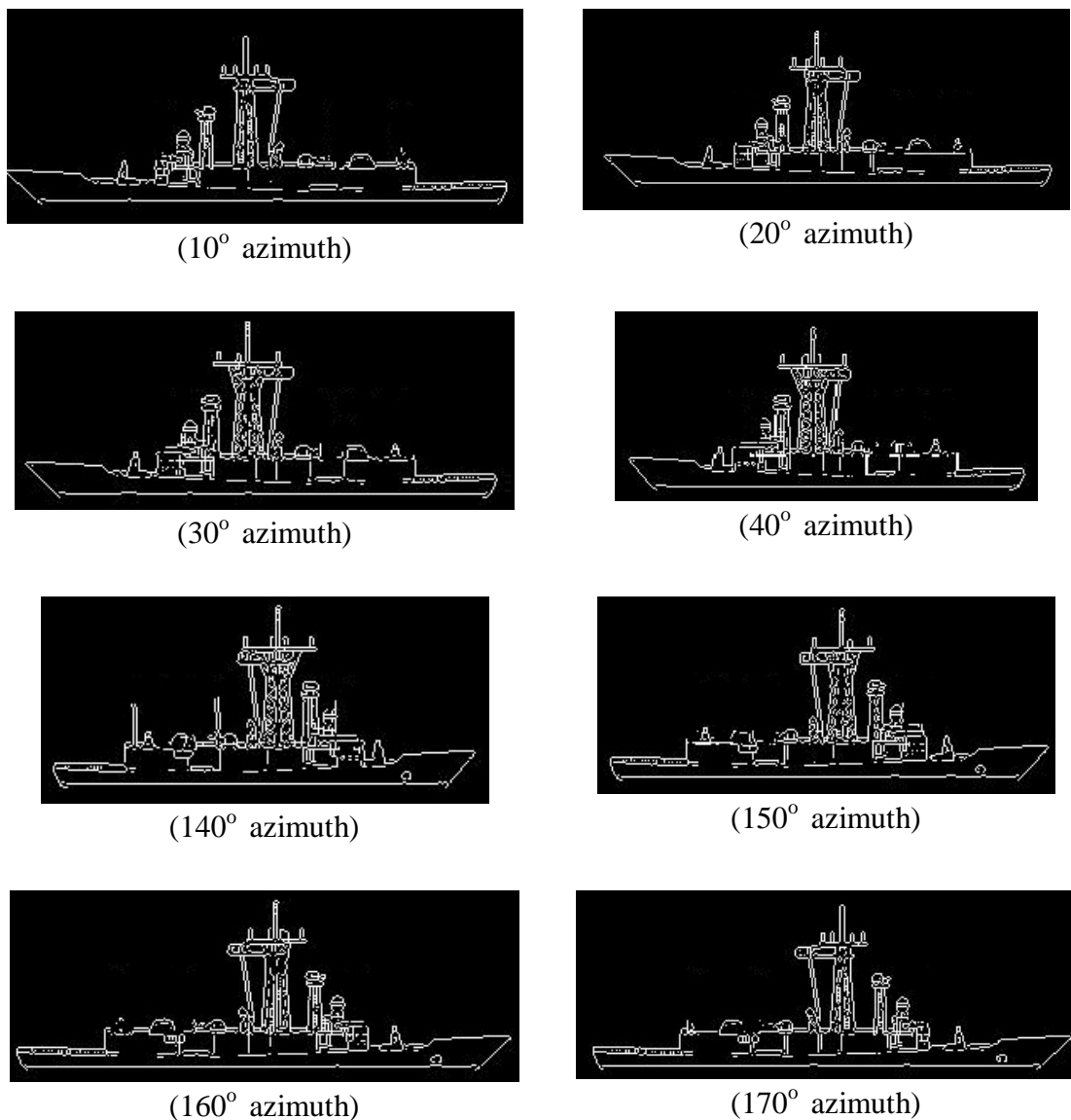


Figure 8: Edge detected synthetic ship images

As seen in the Fig. 8, the results of the synthetic images are so clean and carry more information compared to real images. To make these synthetic images as similar as to real images, Gaussian low pass filter was used to blur the image before using the Sobel edge detector. When the image is blurred enough, some unwanted details disappeared and the edge detected images became more similar to images in real data-sets.

3.2 GrabCut Segmentation Algorithm

GrabCut[26] is an image segmentation method based on graph cuts[8]. This method is proposed for efficient, interactive extraction of foreground object in a complex environment whose background can't be trivially subtracted. It is first starts with hard segmentation by using iterative graph cut. Then, it computes alpha values in the hard segmentation boundary. It is improved version of graph cut algorithm. There are two novel approaches in this method. These are “iterative estimation” and “incomplete labeling” which help to reduce user interaction. This interaction here, just consist of dragging rectangle around the desired object. If the resultant image is not segmented well enough, you can do further enhancement by simply brushing hard segmentation boundary.

But, in this case it is disadvantage for our system. Since, our desire is to build an autonomous ship recognition system. However, in this segmentation method, user should specify the foreground object with dragging rectangle before start to segmentation procedure. Even with the user interaction, it tends to fail for segmentation. Especially, some of the warship images which their colors camouflaged by sea, sky and clouds. These are the results from GrabCut segmentation algorithm.



(Original image)



(Segmented image)



(Original image)



(Segmented image)

Figure 9: GrabCut segmentation algorithm result

As seen in the Fig. 9, the first image is segmented well except the crosspieces of the main mast which are missing, but in the second image, there are some problems in the upper part. GrabCut algorithm is fooled by the sky and the algorithm fails to discriminate superstructure and sky. When the noisier warship image is used in this algorithm, segmentation for the resultant image is getting poorer.

3.3 GrowCut Segmentation Algorithm

This algorithm[29] is proposed for moderately hard segmentation tasks. In the beginning of the algorithm, the user specifies seeds for foreground and background objects. This is done by user's strokes with "foreground" and "background" brushes (like red pixels corresponds to "foreground" brush strokes and blue corresponds to "background" brush strokes). Each paint stroke of a defined brush sets the initial labels

and strength of seed pixels and at each iteration this cell strength is updated. According to our experience, the numbers of the seeds affect the performance of the segmentation. 30 seeds are used for segmentation tasks to get better results in our algorithm. This algorithm generally performs well in most of the images, but like in the GrabCut method, it tends to fail for segmenting some of the warship images which is camouflaged by sea and sky.

This algorithm can be explained with simple intuitive explanation. These seeds can be considered as the bacteria and the bacteria starts to spread from seeds pixels and try to occupy the entire image. The rule of the bacteria is at each discrete time step, it attacks its neighbors. If the bacteria are stronger than the neighboring cell strength, it conquers the cell. The bacteria occupy all the cells at the end and label them is foreground and background according to their strengths. These are the results for GrowCut segmentation algorithm.



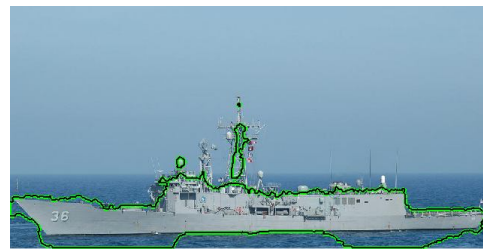
(Original image)



(Segmented image)



(Original image)



(Segmented image)

Figure 10: GrowCut segmentation algorithm results

As seen in the Fig. 10, the segmentation results for the first image is well enough but for the second image, GrowCut algorithm is fooled by noise because of the similar colors in the sea and ship's freeboard. Therefore, this algorithm can't segment the foreground image.

3.4 Proposed Algorithm

All the current good segmentation methods need user interactions to build an autonomous ship recognition system. So, the proposed algorithm is used to make this system autonomous. This algorithm is based on edge detection and enhances the edge detected images to extract useful and robust features. The useful information of the ship images are kept in their superstructure. Because, every ship type has different superstructure which constitutes its uniqueness and we tried to exploit this difference in this system.

For edge detection part, first we tested a powerful method, Canny edge detection, which has advantages over other edge detection methods. Because, Canny method use two thresholds, to detect strong and weak edges, and just display the weak edges in the output if they are connected to strong edges. So, this method is less likely than the others to be fooled by noise and more likely to detect the true weak edges. During the edge detection part with Canny, the default parameters were used as thresholds at first, but the results weren't satisfying. Then, grid search was made for these parameters to find the best thresholds values and it worked for individual images but these thresholds didn't generalize to our overall data-set. The main problem here, every ship image needs different threshold values. Because, waves and clouds are so obvious in some of the images, but in some of the images they aren't. Therefore, choosing hard thresholds for

killing the weak edges which belong to waves and clouds are causing loss of most of the edge information for the superstructure. However, choosing soft thresholds to capture the superstructure information is causing Canny edge detector to be fooled by weak edges. The best possible solution is to find adaptive way to apply Canny method. However, we could not find any good adaptive method[18] for this system. The problem with the parameters selection is illustrated in Fig 11.



(Original image)



(Canny with default parameters)



(Original image)



(Canny with grid searched parameters)

Figure 11: Canny edge detection result

As seen in the Fig. 11, the first ship image is become hardly recognizable because of the noise by wavy sea when the default treshold parameters are used. But, the second image is really good for feature extraction when the grid searched treshold parameters are used. But, as we discussed before, these grid searched parameters are just producing good results for this particular image. If you want to use same parameters for other ship images in this database, the resultant images are became similar to noisy edge detected image in

the Fig. 11. We will stick to this ship image for the later parts to show the enhancement for edge detection.

After Canny edge detection method, we moved to Sobel edge detection which gives better edge detection results for superstructure and it is less fooled by the noise compare to Canny method when the default threshold parameters used in MATLAB®. But, it is not possible to kill all the noise with solely using Sobel edge detection and there is still some noise because of the wavy sea and clouds. Therefore, connected component analysis[5] is applied to kill the remaining noise. Connected components labeling scans an image and groups its pixels into components based on pixel connectivity. For this work, 8 pixels connectivity are used and searched for 15 connected components in the entire image. The idea here is searching for 15 connected components if one pixel belongs to superstructure; it is connected more than 15 pixels. Because, superstructure of the ship contains more connected pixels. So, intuitively if one pixel is connected less than 15 pixels, this particular pixel group can be accepted as noise and we can delete this pixel group. This method works for most of the images contain noise by clouds and it is also help to weaken noise by wavy part of the images. The reason why this method works for particular kind of noise is clouds contain less edge compare to waves.

For the noise by wavy parts of the images, this approach got some help from ship geometry. When the ship images are checked in Fig. 12, it can be easily recognized that there is a distinctive line between sea and ship's superstructure. After this distinctive line, waves follow noisy pattern and most of the pattern are disconnected with each others.



Figure 12: Distinctive line between sea and ship's superstructure

This noise can be made smaller using connected component analysis but we can't kill all of it with only using connected components. So, using Hough transform[28] here in a clever way can help to kill the remaining noise.

In this case, linear Hough transform was used for detecting straight lines. The motivating idea behind the Hough technique for line detection is this algorithm uses an array which called accumulator. The dimension of the accumulator is equal to number of unknown parameters. For linear Hough transform, there are two unknown parameters, which are the pairs of (r, θ) . The two dimensions of the accumulator array correspond to quantized values for (r, θ) . For each pixel and its neighboring pixel, this algorithm determines if there is value of that bin. By finding the bins which have the highest value, and by looking for local maxima in the accumulator, the most likely lines can be extracted. Therefore, all the straight lines can be detected in the image using linear Hough transform. The last line can be accepted as a line between sea and ship's superstructure and we can kill the noise beneath this line by simply converting the white edges into black. This method worked for most of the real ship images in this dataset.

Briefly, this binarization started with Sobel filter. Then, connected component analysis was applied, after that linear Hough transform was used and several binary noise reduction techniques were performed like simple windows to kill the occurrences of isolated noise. Fig. 13 shows each step to contribution for enhancement of the images.

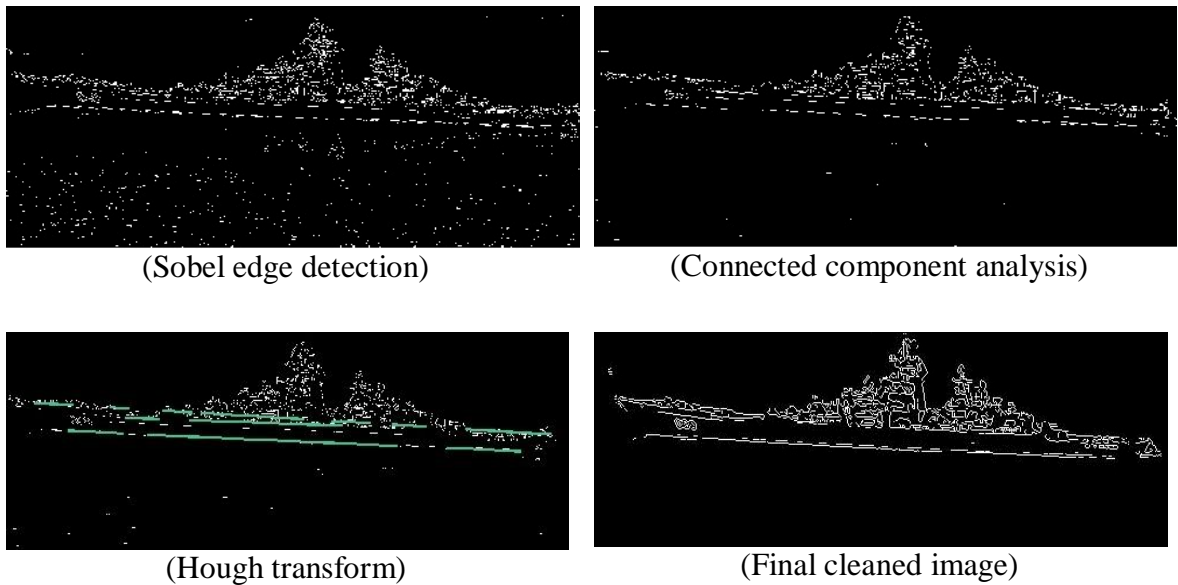
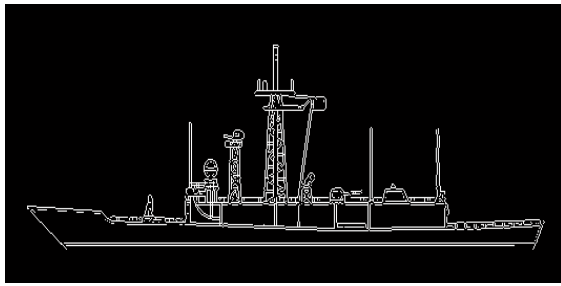


Figure 13: Results for cleaning the edge detected image

The above methods are for the real ship images in our data-sets. Since, synthetic images don't have complex background and noise. So, Sobel filter is enough to detect the edges of foreground object. However, there are some unwanted edges appear in the synthetic images, when we used Sobel edge detector. To kill these unwanted edges, Gaussian low pass filter was used and the resultant procedure is shown Fig. 14.



(Synthetic image)



(Low pass filtered synthetic image)

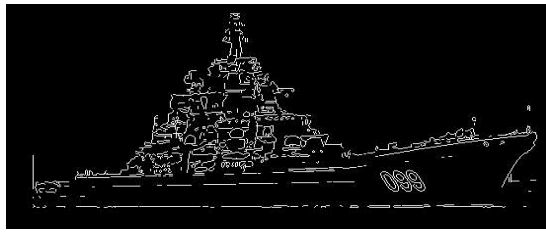


(Real image)

Figure 14: Low pass filtering operation for the synthetic image

As seen in the Fig. 14, after the synthetic image is filtered by Gaussian low pass filter, the resultant image's edge become more aligned with real image's edge.

Fig. 15 shows the sample edge detected real images from different orientation angle which belong to battleship cruiser class in our real data-set. As seen in the figure, as a result of our proposed method, we obtained clean real ship images for our development and testing data-sets.



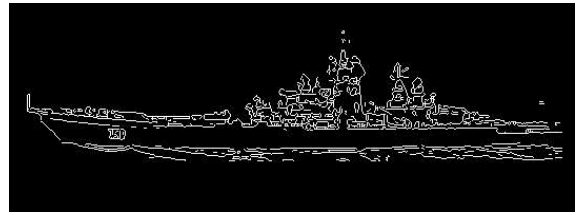
(140 azimuth)



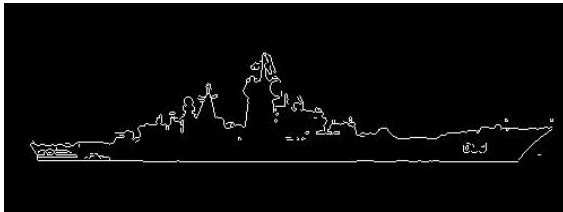
(200 azimuth)



(330 azimuth)



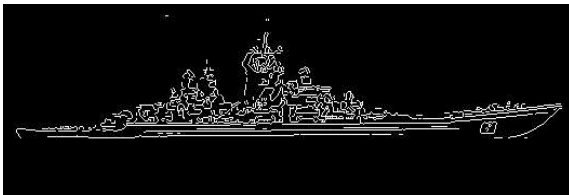
(350 azimuth)



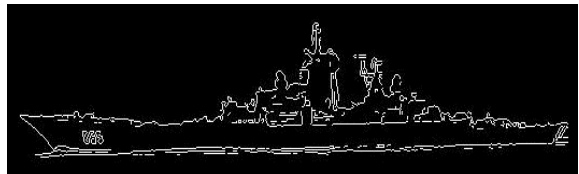
(180 azimuth)



(000 azimuth)



(190 azimuth)



(010 azimuth)

Figure 15: Sample edge detected real images in our database

3.5 Centering and Resizing

After, we binarized our synthetic and real data-sets; there are two important steps to align our feature, which means adjust the position and size of the ship images in this data-set. Since, ships are found at different positions and different sizes in the data-sets. Therefore, synthetic images are used for centering and resizing purposes in this system.

In the real case, we don't need synthetic images for resizing because different ship type covers different space in the screen according to their distance and this distance information (which is coming from RADAR of the ship) can be used as a cue for resizing. But our system is lack of distance information. For this reason, we are using synthetic image's length as a cue instead of distance to resize our real data-set. Fig. 16 is illustrating different positions and sizes of same ship type.



(Ship is close to lower edge of image)



(Ship is close to upper edge of image)



(Ship is large in the screen)



(Ship is small in the screen)

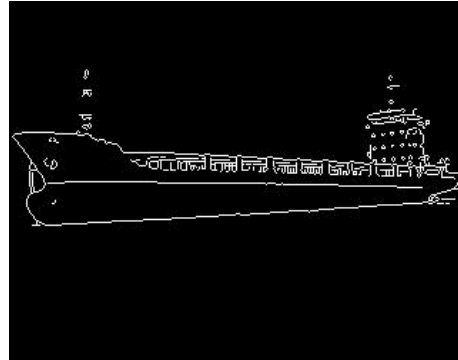
Figure 16: Example for different position and different size of ship images

To be able to resize our real data-set according to synthetic data, centering the images can be a good starting point. In order to center the real ship image, the bounding box is found for the edge detected real ship images. After this step the image is resized according to synthetic image with the same course. Then, "0" values which represent black pixels are padded bounding boxed image until its size became equal to 640 x 640.

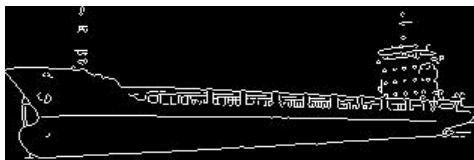
After, completing these steps, the corresponding edges of the same ship type with the same course becomes located at the same position in the edge detected image, which helps to increase the performance of this system.



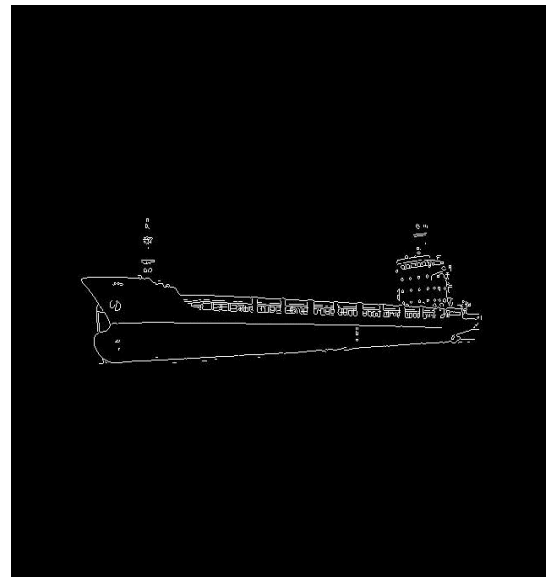
(Original image)



(After segmentation)



(Bounded image)



(Resized and centered image)

Figure 17: Centering and resizing real image in our database

As seen in the Fig. 17, the real ship's image is close to upper size of the picture and the above procedure is used to center and resize the ship image. All the images in our development and testing data-set are resized and centered according to synthetic data in order to align the edges of the images.

Chapter 4

Feature Extraction Model

4.1 Zone Based Feature Extraction Model

This method[25] is widely used to extract features for edge detected images or hand written character recognition. In this method, the images are divided into zones to find edges in the each bin. For this purpose, the numbers of white pixels (which represent edges in the image) are summed in the each bin. 640 x 640 image sizes are used in this system. Actually, the image size seems larger for classification problems but the drawback of using small image in this system is explained in the previous chapter. Therefore, we stick to larger image size.

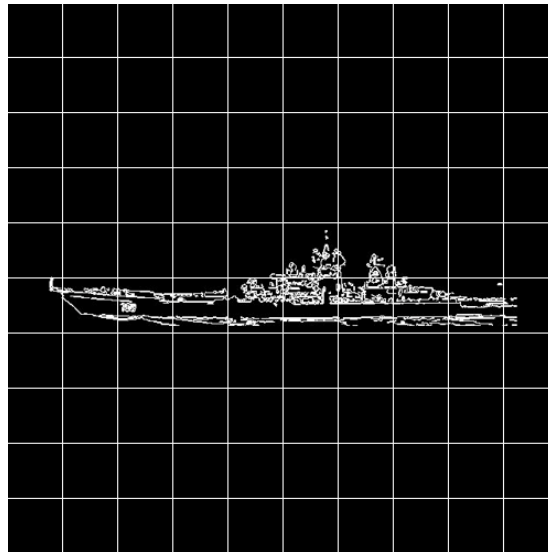


Figure 18: Zone based feature extraction method

For this system, a 10 x 10 window size is selected; the reason is some ship types like fishing or sailing vessels are very small. If larger window size is selected, it would become hard to capture edge variations in the superstructure. Therefore, 10 x 10 window

size is enough to meet our requirements. As seen in the Fig. 18, ship image's length is much larger than its height. So, using square images for this method can produce redundant features which are equal to 0.

Using a 10 x 10 window size for 640 x 640 image make our total feature vector's length 4096 which we will represent as a row vector, but most of the features in the beginning and end part of the feature vector have value 0 as understood from the Fig. 18. Therefore, in order to keep only non-zero values in the feature space and avoid redundant features, all the features in the training vector are summed in a vertical way and result of this summation is a single 1 x 4096 row vector. Then, first non-zero value in the beginning and last non-zero value in the end of the summed feature matrix are found. These non-zero values and the values between them are transferred into another matrix with some margin. The result of this cropping end up with 1100 features vector size. Only these features carry the important edge information for the images. This cropping procedure is illustrated in the Fig. 19.

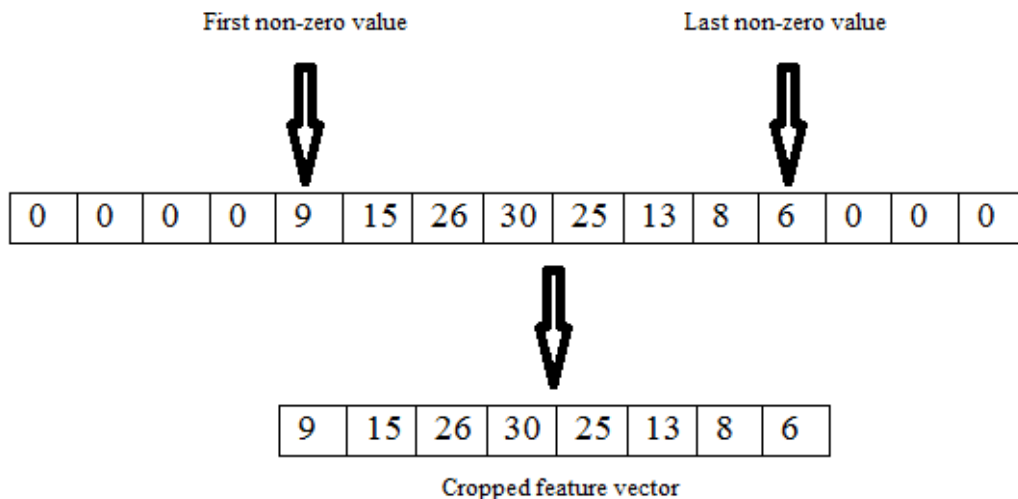


Figure 19: Reducing the feature vector by cropping the zero values

This can be useful for classifiers, because these features are redundant features and cropping these features can increase performance and run-time of our classifier. In order to make sure that this cropping can't affect our performance of the classifier in a negative way, the system is trained and tested with full size feature vectors and separately with cropped feature vectors, and the results of this experiment show that this method of cropping has positive effect on the system.

4.2 Modified Spatial Pyramid Feature Extraction Model

This method[1,7,20], called "Spatial Pyramid Histogram", is produced by partitioning the image into sub-blocks of increasing size and computing histograms of pixels found inside each sub-block. It is generally used for color images to find the color histogram in a hierarchical way. However, the data-sets in this system contain binary images which have pixel values 0 and 1. Here, 0 represents the black background and 1 represents an edge pixel in the image. Therefore, this method must be modified before applying it to these data-sets. For this purpose, projecting every pixel in the column orientation will help us to capture the histogram of the edges. Most of the variations in the ship image are found among different column vectors. Hence, projecting columns give us better results than projecting rows and doing this in a hierarchical way also can help us to increase robustness of this method.

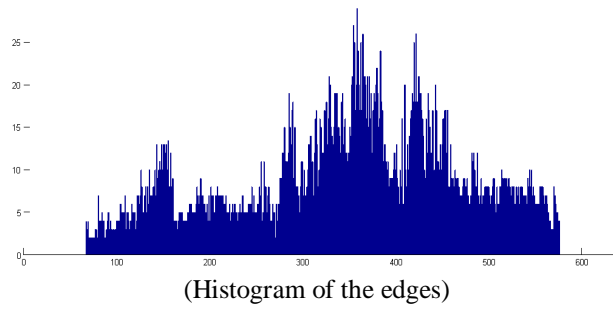
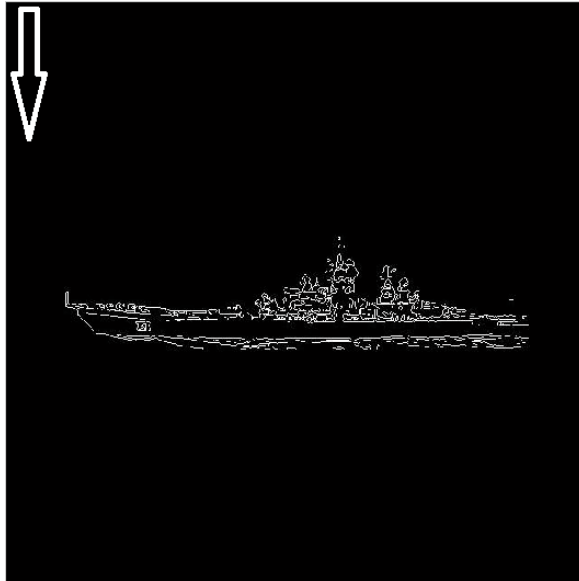


Figure 20: Projecting in the column orientation

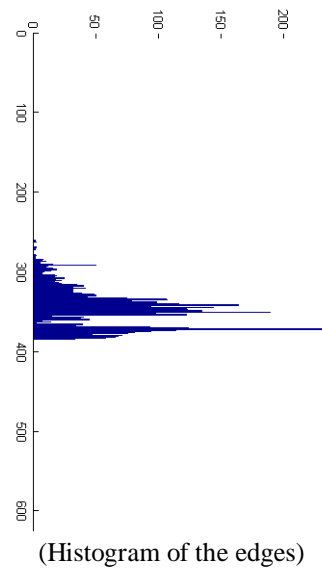
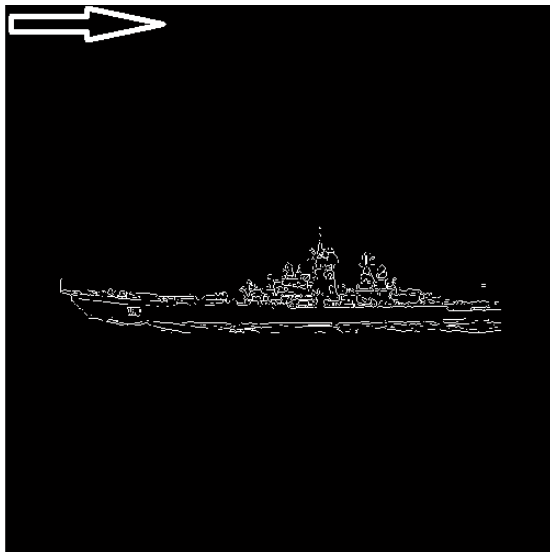


Figure 21: Projecting in the row orientation

As seen in the Fig. 20 and 21, projecting edges gives us better understanding for the image. Projecting in the column orientation gives more broad representation for the superstructure of the ship and projecting in the row orientation gives more narrow representation. Therefore, in this application, projecting column orientation is chosen as a feature extraction method.

This method gave better results in terms of accuracy for the classification compared to zone-based method. One of the reasons, you are representing the image with multi-level representation by using spatial pyramids.

In our modified version, two layers are used for sub-blocks and it is called two layer spatial pyramid histogram and if layer 0 is counted, total number of layers will become equal to 3. The Fig. 22 can show the procedure of how these 3 layers are applied to images in this data-set to capture the features in a hierarchical way.

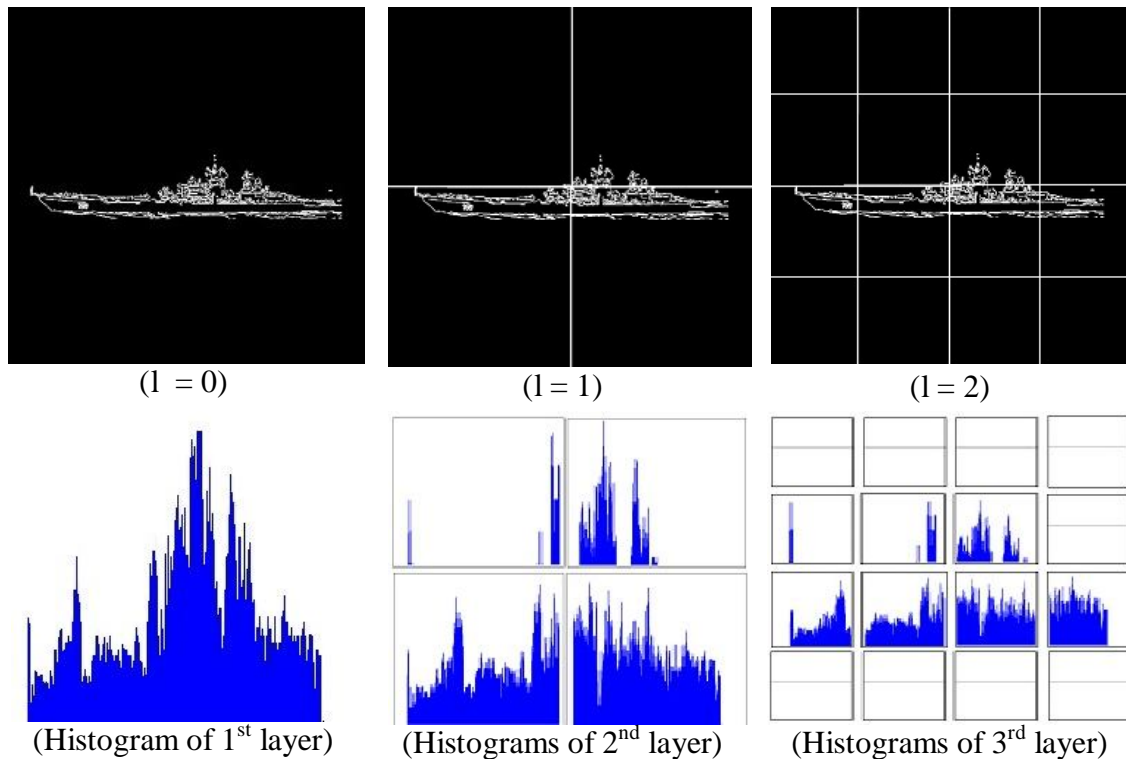


Figure 22: Modified spatial pyramid histogram $L = 2$

In this method, each layer contains the histogram of the edges in the image. For example in the layer 0 in Fig. 22 is the histogram of edges for the binary ship image. This image size is 640 x 640 and when histogram of the edges is found in a column orientation, total number of feature size becomes equal to 640 for the first layer, and for subsequent layers, this feature size keep expanding. The reason is for the second layer the image is divided into four sub-blocks and after that histogram of the edges is found for each sub-blocks. Dividing 640 x 640 images into 4 sub-blocks produce 4 sub-image with size 320 x 320 and taking the histogram of these sub-images produces 4 times 320 features and total features from this layer becomes 1280 features. In the third layer, the same procedure is followed above and continued to divide this sub-image into 4 more sub-image. As a result, 16 images are taken from the last layer. The sizes of these images are 160 x 160 and taking the histogram of these images produce 16 x 160 features and total number of the feature vector size becomes 2560. At the end, total number of these features from each layer are concatenated to find final feature vector size and it becomes 4480 features for one binary image.

Before we start the classification part, we are expecting better results from spatial pyramids method; the reason is, it keeps track of the edges in a hierarchical way, and taking projection of the edges should give better understanding than the zone based method.

Chapter 5

Feature Reduction and Feature Selection

5.1 Dimensionality Reduction

Dimensionality reduction[13] refers to the process of taking a data-set with a high dimension, and then creating a new data-set with lower dimension, which is meant to capture only those dimensions that are in some sense “crucial”. The main purpose of dimension reduction is that we want to preserve as much “structure” of the data as possible, while reducing the number of dimensions it has; so, we are creating a lower dimensional approximation to the data. This dimension reduction has several benefits for us as follows:

- (1) It reduces the run-time of our classifier, because for many applications, run-time is important for the user, so using lower dimension can cause a faster convergence for the classifier.
- (2) Sometimes, it increases the accuracy, because reducing the number of dimensions also reduces redundancy of features in the data-set.
- (3) Since the structure is largely preserved, we can apply this reliable approximation for our original testing data.

For the remainder of this chapter, several dimension reduction methods are explained that are used for this application. We also discuss feature selection methods[5] which are used in two different versions for this system. There will be a brief explanation

about differences between dimension reduction using transformation and feature selection.

5.2 PCA

Principal Component Analysis (PCA)[27] is a very popular method for dimensionality reduction that uses a very simple idea. Suppose we have a high-dimensional data-set. The dimensions can be thought of as directions that the data varies along.

If the dimension of the data wants to be reduced, we should find an approximate basis to the set of directions. Only important dimensions want to be found. Similarly, our aim is to find dimensions which have the maximal variance. The points are approximately constant along the other dimensions that are in some sense less important. This is the general intuition how the PCA method works in theory.

If how the PCA method works in MATLAB® wants to be discussed, “princomp” command should be introduced. When the princomp command is used in MATLAB®, “[COEFF,SCORE,latent] = princomp(A)”. It gives us the eigenvalues of the covariance matrix A (which is latent) in decreasing order. In most of the application for feature reduction with PCA, 95% of the latent capture all the variance in the data matrix and it determines our feature vector size for reduction.

$$latent^t = [e_1 e_2 e_3 \dots e_n], \quad e_1 > e_2 > e_3 > \dots > e_n, \quad e_i \text{ is eigenvalue } i$$

$$k = \frac{\sum_{i=1}^{i=j} e_i}{\sum_{i=1}^{i=n} e_i}, \quad j = 1, 2, 3, \dots, n \quad \text{and } k > 0.95$$

In the summation equation, j variable starts with 1 and will be increased until k becomes greater than 0.95. When k becomes greater than 0.95, we will stop increasing j and keep that specific j value. This j value will be our feature size for feature reduction. For example, suppose n equals to 500 and when j is equal to 100, k becomes greater than 0.95. Then, it means that 100 features have the maximal variance in this data-set.

PCA was performed in this system and it gave us poor results for classification. Our total feature size is 4480 and we reduced it down to 200. The “princomp” command is used in MATLAB® in order to find the reduced feature size and in the latent of our data matrix, k becomes greater than 0.95 when j equals to 200. Using 200 features from PCA reduced our accuracy for classification in a considerable amount. So, other feature extraction methods are used instead of PCA for this system.

5.3 Random Projection

Briefly, random projections[6] involve taking a high dimensional data-set and mapping it into lower-dimensional space. This method provides some guarantees on the approximate preservation of distance. How random projection works can be explained as follows. Suppose the input data is an $n \times d$ matrix A , representing the n feature vectors, each of dimension d . Then, to do projection, a suitable $d \times k$ matrix R has to be selected, and then projection can be defined as

$$E = AR$$

Now, dimension of A is reduced to $n \times k$.

Random projection can be done using the Johnson-Lindenstrauss lemma, which tells us the following.

THEOREM 1. (Johnson and Lindenstrauss, 1984) [11] Suppose we have an arbitrary matrix $A \in \mathbb{R}^{n \times d}$. Given $\varepsilon > 0$, there is a mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^k$, for any $k \geq 12 \frac{\log n}{\varepsilon^2}$, such that, for any two rows $u, v \in A$, we have

$$\ell(y, \hat{y}) = 1, \text{ if } y \neq \hat{y}, \text{ and } 0 \text{ otherwise}$$

The lemma says that if we reduce to $k = O(\log n / \varepsilon^2)$ dimensions, we can preserve pair wise distances up to factor of $(1 \pm \varepsilon)$.

This lemma tells us about existence of such a mapping but doesn't say how we can actually find one. For this purpose, one more theorem will be given to find one such a mapping. Achlioptas, provided a way to find a mapping in a constant time that provides the above guarantees, but with some explicit probability. His result was the following.

THEOREM 2. (Achlioptas) [2] Suppose that A is an $n \times d$ matrix of n points in \mathbb{R}^d . Fix constants $\varepsilon, \beta > 0$, and choose an integer k such that

$$k \geq k_0 := \frac{4 + 2\beta}{\varepsilon^2 / 2 - \varepsilon^3 / 3} \log n$$

Suppose now that R is a $k \times d$ matrix with entries r_{ij} belonging to the distribution

$$r_{ij} = \sqrt{3} \cdot \begin{cases} +1 & p = 1/6 \\ 0 & p = 2/3 \\ -1 & p = 1/6 \end{cases}$$

Define the $n \times k$ matrix $E = \frac{1}{\sqrt{k}} AR$, which we use as a projection of A onto a k -

dimensional subspace. For any row u in A , let us write $f(u)$ as the corresponding row in E . Then, for any distinct rows u, v of A , we have

$$(1 - \varepsilon) \|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon) \|u - v\|^2$$

with probability at least $1 - n^{-\beta}$.

Achlioptas says that we can control the parameters ϵ, β to get, say 99.99% chance of no distortion.

There are several reasons why random projection has proven to be a popular technique for dimensionality reduction. Firstly, unlike other methods of dimensionality reduction, such as PCA, random projections are quite easy to do, as they just require a matrix multiplication with a matrix that can be generated very simply. You don't need to find any covariance matrix like in PCA and other methods.

Another property of random projections that is appealing, they manage to approximately preserve individual lengths, and pair wise distances – specifically, they approximately preserve the length of the original points, as well as the distances between original points with arbitrary accuracy.

Clearly, the choice of projection matrix is crucial in determining if the random projections have their distance-preserving properties. Indeed, the term “random” in random projections should not be taken to mean that we are projecting onto some completely arbitrary subspace. It is better to think of the situation as having some underlying structure, but within this structure very loose restriction.

In our application we picked Achlioptas matrix which is simple to apply, if we briefly describe this matrix:

$$r_{ij} = +\sqrt{3} \begin{cases} +1 & p = 1/6, \\ 0 & p = 2/3, \\ -1 & p = 1/6 \end{cases}$$

Achlioptas matrix is very easy to compute, requiring only a uniform random generator, but even more importantly, it is two-thirds sparse, meaning that essentially two-thirds of our work can be cut off.

In this system, random projection method is successfully applied. However, There is a problem with generation of the random matrix R. “rand” command used in MATLAB® generates random numbers which is giving values between (0,1). But, because it is a random number generator, it gives different numbers in the matrix each time. So, when we get the best accuracy for projecting onto lower-dimension, the same matrix can’t be obtained using rand command anymore because it is changing each time. For this purpose, we played the seeds of random number generators in MATLAB® in order to fix the numbers in the projection matrix. In this system, rand command gives us the same matrix for each time, so that this same matrix can be used for test data-set.

5.4 Sequential Forward Selection (SFS)

Feature selection[19] is different from feature reduction in some sense. Feature reduction transforms the existing features into lower dimensional space however feature selection selects a subset of the existing features without transformation. So SFS looks at the issue for dimensionality reduction from a different perspective.

Feature selection

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iM} \end{bmatrix}$$

Feature reduction

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = f \left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \right)$$

Figure 23: Feature selection

Fig. 23 shows how these two algorithms work. Feature selection is typically a search problem for finding the optimal or suboptimal subset of m features out of original M features. Feature selection is important in many pattern recognition problems for excluding irrelevant and redundant features. It allows reducing system complexity, run-time and often improves the recognition accuracy.

SFS starts from the empty set, and sequentially adds the features that maximize the classification accuracy when combined with the features that have already been selected. In more detail, algorithm starts by picking the highest accuracy feature from overall data-set. After that keeping this feature and combine this with the new selected features one at a time. SFS algorithm is simple to implement and fast; they have time complexity of $O(M^2)$ (where M is the number of features) or less. However, as they don't perform complete search, they may miss the optimal feature subset.

There are more advanced methods in feature selection like Sequential Floating Forward Selection (SFFS). In this case, locally best features are selected out of overall features, and they are combined sequentially with the newly selected ones but sometimes this combination is not optimal. SFFS procedure consists of applying after each forward step a number of backward steps as long as the corresponding subsets are better than the previously evaluated ones at that level. But, this procedure increases the complexity and run-time of the algorithm substantially.

SFS algorithm is used in two different cases in this system. In the first case, we used random projection to reduce the number of features to 450, and after that the best 300 feature subset is selected out of overall 450 features using SFS. The idea here, after feature reduction with random projection method, we are transforming our features to

lower dimensional subspace. However, after this transformation, there can be still redundant or irrelevant features. Therefore, using SFS may help to eliminate these redundant features and increase our accuracy and run-time. In the second case, SFS is used for our overall data-set and the best 600 features are selected out of 4480 features. The main problem in the second case, our feature dimension is high; it takes approximately 3-4 days to choose the best 600 features. Since, SVM classifier is performed for each possible newly selected feature. For that reason, SFS algorithm is used instead of SFFS because the complexity in SFFS is much higher than the SFS.

Chapter 6

Classifier and Classification Results

6.1 Support Vector Machine (SVM)

Support Vector Machines (SVM) [9,16] provide a useful technique for data classification. The method is easy to use and suitable for large data-sets. SVM perform classification by constructing an N-dimensional hyperplane that optimally separates the data. This classification method is based on the concept of decision planes that define decision boundaries. A decision boundary separates objects that belong to different classes. SVM can be used for multiclass classification by building binary classifiers which distinguish between one of the labels and the rest (one-versus-all) or between every pair of classes (one-versus-one). The classification for the one-versus-all case is done by a winner-takes-all strategy, for the one-versus-one case is done by a max-wins voting strategy. A good separation is achieved by the hyperplane that has the largest distance to the nearest object of any class, because in general the larger the margin the lower the error of the classifier.

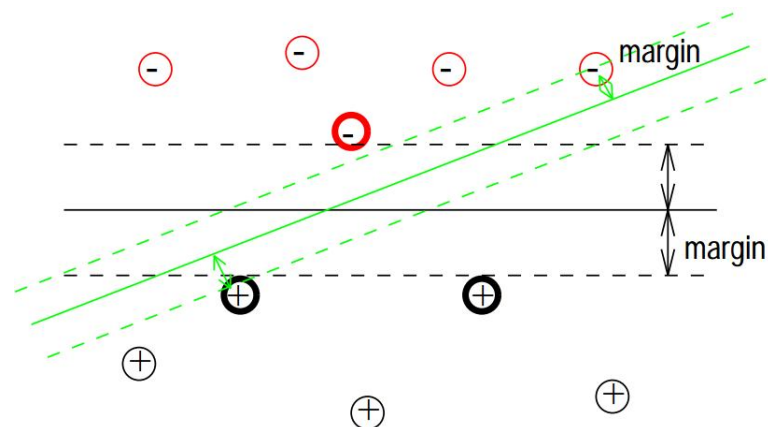


Figure 24: Support vector machine for separable case

We want to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible while still separating the data. In Fig. 24 the training data is linearly separable; we can choose the two hyperplanes of the margin in a way that there are no objects between them and try to maximize distance between these hyperplanes.

In Fig. 24, the support vectors are indicated by the circles around them. These are the critical elements of the training set. They lie closest to decision boundary; if all the other training objects were removed and training procedure was repeated, the same separating hyperplane would be found. To find the maximum margin separator which is the function of the support vectors, we have to solve the following optimization problem for the linearly separable case:

Our criterion function can be written:

$$\frac{1}{m} \sum_{i=1}^m \ell(f(x_i, \alpha), y_i) + \|w^2\|$$

ℓ is the zero-one loss function:

$$\ell(y, \hat{y}) = 1, \text{ if } y \neq \hat{y}, \text{ and } 0 \text{ otherwise.}$$

$f(x, \alpha)$ is a function with parameter α .

Minimize $\|w^2\|$ with respect to w , subject to constraints:

$$\begin{aligned} (w \cdot x_i + b) &\geq 1, \text{ if } y_i = 1 \\ (w \cdot x_i + b) &\leq -1, \text{ if } y_i = -1 \end{aligned}$$

This is the tricky but convex problem. There is only one optimum and we can find it without fiddling learning rates or weight decay or early stopping.

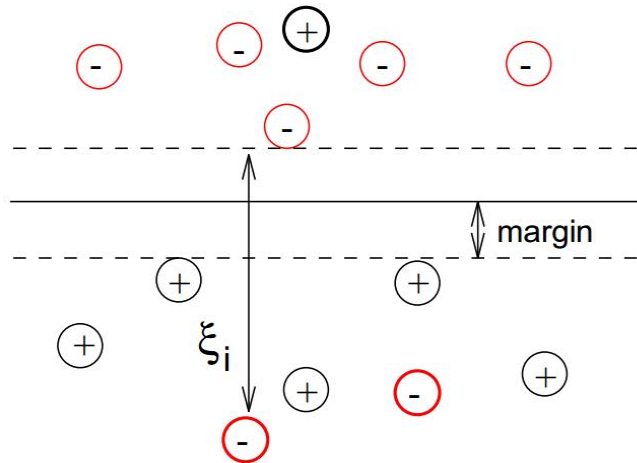


Figure 25: Support vector machine for non-separable case

To deal with the non-separable case, we have to introduce slack variables and the problem can be rewritten as:

Minimize $\|w\|^2 + C \sum_{i=1}^m \xi_i$ with respect to w , subject to constraints:

$$y_i(w \cdot x_i + b, y_i) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

ℓ is no longer the zero-one loss, but is called the “hinge-loss”.

Linear classifiers aren't complex enough sometimes and non-linear classifiers can be necessary to classify our data-set. For this reason, our data-set can be mapped into a richer feature space including non-linear features, then construct a hyperplane in that space so all other equations remain the same.

First mapping our features:

$$x \rightarrow \Phi(x)$$

Then learn the map from $\Phi(x)$ to y :

$$f(x) = w \cdot \Phi(x) + b$$

SVM kernel function:

$$K(x_i, x_j) = \varphi^T(x_i)\varphi(x_j)$$

The following figure helps us to understand how non-linear mapping works to classify our data-set.

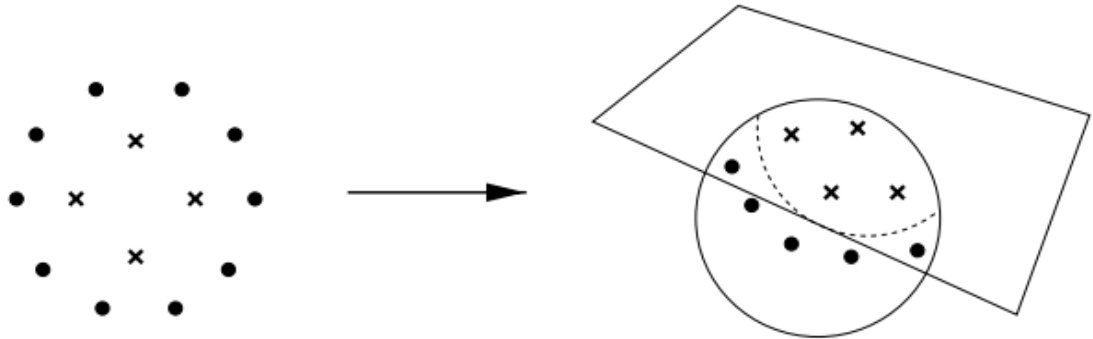


Figure 26: Non-linear mapping for SVM classifier

For these data-sets a library for support vector machines (libsvm) is used which is trusted tool for SVM classification. This tool offers the following kernels:

- linear: $K(x_i, x_j) = x_i^T x_j$.
- polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$.
- radial basis function(RBF): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$.
- sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.

Here, γ, r , and d are kernel parameters.

We have 9 classes in these data-sets; therefore multi-class linear SVM classifier was used in this system. In the libsvm, the above kernels were tried with kernel parameters, and the best result is obtained by using RBF kernel. After that, to enhance our accuracy, the grid search was made for cost parameter “ c ” and RBF parameter “ g ”

(γ parameter used as g in the libsvm) to find the optimal values. The optimal parameter values for the spatial pyramid histogram are $c = 1$, $g = 0.001$ and for the zone based method are $c = 10$, $g = 0.0005$.

6.2 Classification Results for Training Data

In this part, we will present our results for the training set which is synthetic data-set versus development data-set. The parameters were obtained using training on synthetic data-set and testing on development data-set. Cross-validation wasn't used in this part. We did all these experiments in this section for the development set to make this work more realistic. After, searching for various methods like feature extraction, feature combination, feature reduction, best results are presented for this training set. Different classifiers are used in Pattern Recognition Toolbox (prtools) [24] and "libsvm"[10] and the best accuracy is obtained by using SVM. In the first part, SVM classifier is trained on synthetic data-set and tested on development data-set to set up the cost parameter c and RBF kernel parameter g . The accuracy and run-time of SVM classifier will be presented in this classification part. The results in Table 1 are just based on SVM classifier on the "libsvm" tool.

Feature Extraction	Feature Size	Classifier	Accuracy (%)	Run-time (sec.)
Zone based	1100	SVM	69.44	0.28
Spatial pyramid	4480	SVM	77.77	1.01

Table 1: Results for training set vs. development set

If we make evaluation on the results in Table 1, the spatial pyramids result is much better than zone-based result, and this is what was expected at the beginning. Spatial pyramid is taking the histogram of the image. Therefore, much meaningful information can be obtained in a hierarchical way. In the zone-based method, just taking

the edge information in the each bin is giving us limited information about the image. However, run-time of the zone-based method is much faster than the spatial pyramids. Because, we cropped the zero parts of the feature vector for the zone-based method and the feature vector size decreased to 1100, which is approximately a quarter of the size of the spatial pyramid feature vector. From now on, the results for feature reduction and feature selection method will be presented for spatial pyramids feature vectors.

In Fig. 26, feature reduction method is used which is based on random projection; starting from 4480 features, it is reduced down to 1 feature by decreasing 1 feature at a time. But, in order to make it more clear, the following graph will just show results only for each multiple of 100 features. Best result for accuracy is found in the 457th feature.

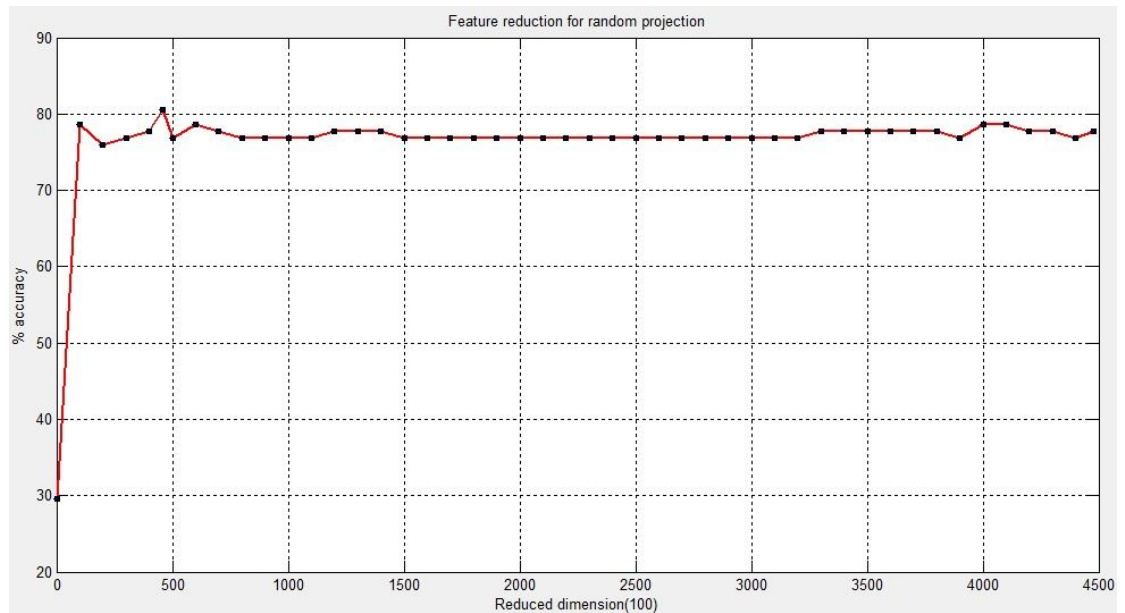


Figure 27: Feature reduction (Random projection)

For random projection, good results are obtained for this training set for more than approximately 100 features, showing that our accuracies are almost equal to original results with the overall features and run-time is reduced substantially as seen in Fig. 27. One of the reason is our feature matrix contains many zeros and thus is a sparse matrix,

and random projection method is a recommended method for feature vectors that are sparse.

Feature Extraction	Feature Size	Classifier	Accuracy (%)	Run-time (sec.)
Spatial pyramid	457	SVM	80.55	0.24

Table 2: Result for random projection

After performed the feature reduction with Random Projection, feature selection method is applied to transformed features. The best 300 features are selected out of 457. Result of this feature selection is shown in Table 3.

Feature Extraction	Feature Size	Classifier	Accuracy (%)	Run-time (sec.)
Spatial pyramid	300	SVM	78.70	0.21

Table 3: Result for random projection + feature selection

As seen in Table 3, to apply feature selection to transformed features did not improve our results, it just has little effect on our run-time.

In Fig. 27, we applied SFS and the best 600 features are selected out of 4480 features using only SFS. Result of this feature selection is shown in Fig. 23.

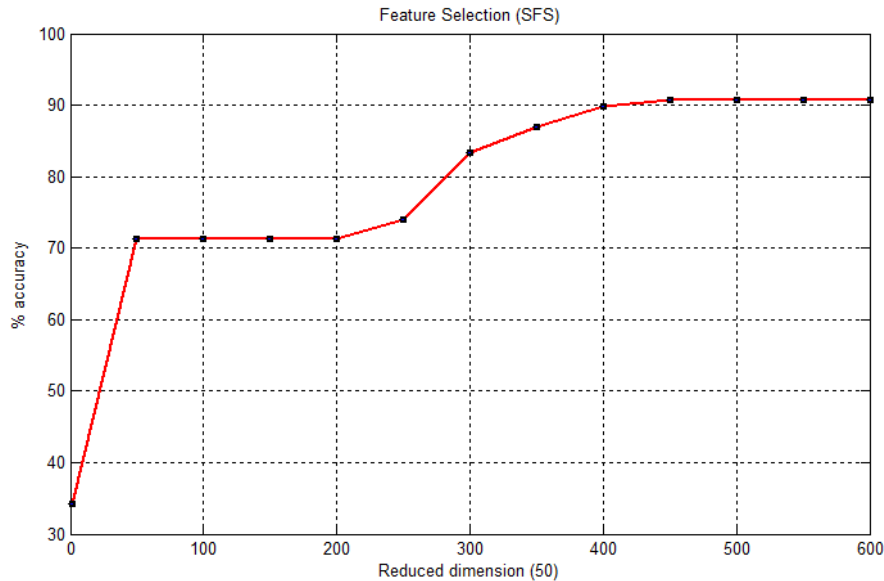


Figure 28: Feature reduction (SFS)

As seen in Fig 28, accuracy results are getting better after 600 features are selected. In order to make the graph more understandable, just each multiples of 50 features are plotted.

Feature Extraction	Feature Size	Classifier	Accuracy (%)	Run-time (sec.)
Spatial pyramid	600	SVM	90.74	0.27

Table 4: Result for feature selection

The best accuracy for the training set is taken from feature selection. However, to choose 600 features out of 4480 feature took 3-4 days and when the selected features are checked, it didn't choose any features from 3rd layer in our spatial pyramid method. So, to make this method robust, more features have to be selected.

In the last experiment, the synthetic data and real data are combined in our training set and we did 10-fold cross-validation to select the best cost "c" parameter and RBF parameter "g" in libsvm. The best results are obtained when $c = 1$ and $g = 0.06$. After that, 10-fold cross-validation is performed for 10 times and the mean of the result is taken.

Feature Extraction	Feature Size	Classifier	Accuracy (%)
Spatial pyramid	4480	SVM	95.83

Table 5: Result for 10-fold cross-validation

As we expected, when these data-sets are combined, higher accuracy is taken for the cross-validation. The main reason is accuracy for the synthetic data is close to 100% and it helps to increase accuracy for overall data-set. Feature reduction and feature selection methods are not performed for the combined data-set.

6.3 Classification Results for Testing Data

In the section 6.2, we trained SVM classifier on synthetic data-set and tested on development data-set to obtain g and c parameters for the SVM classifier and also to perform feature extraction and feature reduction methods. Cross-validation was not used in this part. Then once the values of g and c were found and best feature extraction and feature reduction methods were determined, the SVM was trained on only synthetic data, and tested on the testing data-set; this was done for 3 different cases of features sets as shown in the Table 6.

These three results for our testing data are one for overall features, one for random projection and one for feature selection. After that, the confusion matrix will be given for testing data.

Feature Extraction	Feature Size	Classifier	Accuracy (%)	Run-time (sec.)
SP	4480	SVM	85.56	0.9691
SP + RP	457	SVM	87.78	0.26
SP + SFS	600	SVM	83.33	0.27

Table 6: Results for training on synthetic data-set only; testing on test data-set

In the second part, we will present our result for the combined data-set. In this experiment, SVM parameters g and c were tuned using synthetic data-set + development data-set, with 10-fold cross-validation; this was done using the full feature set (4480 features). These values of g and c were then fixed and used in the following experiment. An SVM was trained on the synthetic data-set + real data of development data-set and the resulting classifier was tested on the testing data; this was done only for full feature set (4480 features), with result shown in Table 7.

Feature Extraction	Feature Size	Classifier	Accuracy (%)	Run-time (sec.)
SP	4480	SVM	93.33	1.46

Table 7: Results for training on combined (synthetic and real) data-set; testing on test data-set

The result is better than the previous three results in Table 6 as expected. The main reason is here when synthetic data-set is combined with the real data-set; it better tunes the decision boundary in our SVM classifier. Therefore it gave us better results. The numbers of the synthetic and real images are not the same in our combined data-set. If we increase the number of real images in the combined data-set, the accuracy should keep increasing.

Inferred Type / Input Type	Cruise	Fishing vessel	Battleship cruiser	Container vessel	Corvette	Frigate	Sailing vessel	Oil tanker	Cruiser warship	Recall
Cruise	10	0	0	0	0	0	0	0	0	100%
Fishing vessel	0	10	0	0	0	0	0	0	0	100%
Battleship cruiser	1	0	9	0	0	0	0	0	0	90%
Container vessel	0	0	0	9	0	0	0	1	0	90%
Corvette	0	0	0	0	10	0	0	0	0	100%
Frigate	0	0	0	0	0	5	0	0	5	50%
Sailing vessel	0	1	0	0	0	0	9	0	0	90%
Oil Tanker	3	0	0	0	0	0	0	7	0	70%
Cruiser warship	0	0	0	0	0	0	0	0	10	100%
Precision	71.4%	90.9%	100%	100%	100%	100	100%	87.5%	66.7%	
Overall probability of classification										87.78%

Table 8: Confusion matrix for training on synthetic data-set only; testing on test data

Inferred Type / Input Type	Cruise	Fishing vessel	Battleship cruiser	Container vessel	Corvette	Frigate	Sailing vessel	Oil tanker	Cruiser warship	Recall
Cruise	10	0	0	0	0	0	0	0	0	100%
Fishing vessel	0	10	0	0	0	0	0	0	0	100%
Battleship cruiser	0	0	10	0	0	0	0	0	0	100%
Container vessel	0	0	0	10	0	0	0	0	0	100%
Corvette	0	0	0	0	10	0	0	0	0	100%
Frigate	0	0	1	0	0	7	0	0	2	70%
Sailing vessel	0	0	0	0	0	0	10	0	0	100%
Oil Tanker	1	0	1	0	0	0	0	8	0	80%
Cruiser warship	1	0	0	0	0	0	0	0	9	90%
Precision	83.3%	100%	83.3%	100%	100%	100%	100%	100%	81.8%	
Overall probability of classification										93.3%

Table 9: Confusion matrix for training on combined (synthetic and real) data-set; testing on test data-set

6.4 Comparison of Results with Similar Works

There are no other works in the literature that are directly comparable, but there are similar ones. The most similar is published in a thesis from NPGS[3] (Naval Post Graduate School) which is “Recognition of Ship Types from An Infrared image”. Infrared images are little hard to work with and it is hard to detect edges. Instead of edges, it is better to use silhouette of the images. However, other segmentation, classification, feature reduction and feature selection problems are similar. Therefore, we can compare our results with their work. When their results are checked, they just used 5 ship types and 25 images for their real data-set and images are not shared equally among the ship types. Their overall accuracy is 68% for real ship Forward Looking Infrared (FLIR) images.

In our work, we used 9 ship types, our test data-set has 90 images and they are shared equally among the ship types. This is one of the important points, because if you use 2 images for a poorly classified ship type and 20 images for an easily classified ship type, your final result doesn't reflect the truth. Our testing results are 83.33% - 87.78% accuracy for synthetic vs. real case and 93.33% for synthetic + real vs. real case.

The differences for the accuracies between two works arise from following reasons. First of all, IR images are hard to work with compare to color images. However, our segmentation technique is different from theirs. We used spatial pyramids which help us to increase our accuracy and random projection produced better results for feature reduction. In addition, our 3D models are built in a more professional way and these contributions help us to improve our accuracy.

Chapter 7

Conclusion

7.1 Concluding Remarks

In this work, we tried to identify unknown ship types from their side views within the limitation of $\pm 40^\circ$. Our approach started with collecting data for both synthetic and real ship images. After that, segmentation part took most of our time. We tried a variety of methods for segmentation, and we also tried feature extraction methods which don't need segmentation. Overall, half of our time was spent on converting color ship images into a meaningful format for feature extraction. For the feature extraction part, previous works used zone-based method and similar ones, but we came up with new solution for this part and this is one of our biggest contributions to this work. Also, we showed that using random projection for feature reduction helps to speed up feature reduction process for large feature vectors. As a result, we obtained 93.3% accuracy for the ship recognition problem.

At the end, we mostly achieved our goals. For classification part synthetic vs. real, we got poor classification results for only the frigate. The main reason for this poor classification is because of the synthetic model that we used. In this part, synthetic model for frigate is not reflecting the real ship image, even with the Gaussian low pass filter; we can't remove the unwanted details for superstructure of the frigate. Therefore, this warship class confused with other warship class as seen in the confusion matrix.

7.2 Future Works

This work can be further extended in the following directions.

1. More ship types / classes: In this work, we stick to 9 ship types but actually there are many more ship types, and the ship types can be divided into ship subclasses to provide more specific identification for military application. For example, we can detect the ship as a frigate, but there are approximately 30 different subclasses of frigate and each class is used by a different country. This subclass identification can be very important for military intelligence.
2. All perspectives: In this work, we used a limited range of perspective which is closed to side view because it is hard to recognize ships from front views, and we had a very limited number of real images of frontal views of ships. Also, 10 degree resolution was used. All the views of the ship with fine resolution can be used to make this system more effective for the real case.
3. Infrared images: Most of the warships have FLIR cameras nowadays. Therefore, this work can be extended for FLIR images. Because during the night, color cameras become useless and the ships are detected over FLIR cameras. Therefore, 24 hours detection becomes available for the ship recognition system. In addition, we can combine the results for FLIR and color images to enhance our detection result. Unfortunately, FLIR images couldn't be used for this system because it is hard to find publicly available IR ship images for different ship classes.
4. Segmentation: Using segmentation before edge detection can enhance the final results for segmented images. If we can segment the color ship images with fully automated segmentation methods, it really helps to detect ships with certain

background. With the current method, it is really hard to segment ship images which have distinctive background like city or mountain.

5. Course information: With current method, we did not use course information. However, using course information can give us better results. For example, comparing 90° course ship with other 90° course ships in the training set will help us to increase our accuracy. Course and distance information is available from RADAR of the ship. So, using all the available information in the ship can enhance our results.
6. SIFT features: Scale Invariant Feature Transforms (SIFT) image features provide a set of features of an image that are not affected by many of the complications experienced in other methods, like object scaling and rotation. So, in this data-set SIFT features can help to skip resizing and centering part for the real images and can enhance the accuracy.

References

- [1] A. Abdullah, R. C. Veltkamp, and M. a. Wiering, Spatial pyramids and two-layer stacking SVM classifiers for image categorization: A comparative study, *2009 International Joint Conference on Neural Networks*, pp. 5-12, Jun. 2009.
- [2] D. Achlioptas, Database-friendly random projections: Johnson-Lindenstrauss with binary coins, *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671-687, 2003.
- [3] J. Alves, J. Herman, N. C. Rowe, Robust Recognition of Ship Types from an Infrared Silhouette, Naval Post Graduate School, Monterey, 2004.
- [4] J.A. Alves, Recognition of ship types from an infrared image using moment invariants and neural networks, Naval Post Graduate School, Monterey, 2001.
- [5] D. G. Bailey and C.T. Johnston, Single Pass Connected Components Analysis, *Image and Vision Computing New Zealand*, vol. 10, 2007.
- [6] E. Bingham and H. Mannila, Random projection in dimensionality reduction : Applications to image and text data, *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 245-250, 2001.
- [7] A. Bosch, A. Zisserman, and X. Munoz, Representing shape with a spatial pyramid kernel, *Proceedings of the 6th ACM international conference on Image and video retrieval*, pp. 401-408, 2007.
- [8] Y. Y. Boykov and M.-P. Jolly, Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images, *In Proc. IEEE Int. Conf. on Computer Vision*, July, pp. 105-112, 2001.
- [9] C. J. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data mining and knowledge discovery*, vol. 2, no.2, pp. 161-167, 1998.
- [10] C.-Chung Chang and C.-Jen Lin, LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.
- [11] S. Dasgupta and A. Gupta, An elementary proof of a theorem of Johnson and Lindenstrauss, *Random Structures and Algorithms*, vol. 22, no. 1, pp. 60-65, 2003.
- [12] R.O. Duda and P. E. Hart, Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Communication of the ACM*, vol. 15, no. 1, pp. 11-15, 1972.

- [13] I. K. Fodor, A survey of dimension reduction techniques, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory ,vol. 9, pp. 1-18, 2002.
- [14] V. Gouaillier and L. Gagnon, Ship Silhouette Recognition Using Principal Components Analysis, Proc. SPIE Conf. on Applications of Digital Image Processing XX, vol. 3164, pp. 59-69, 1997.
- [15] J. Herman, Target Identification Algorithm for the AN/ASS-44V Forward Looking Infrared (FLIR) , Naval Post Graduate School, Monterey, 2000.
- [16] C.-Wei Hsu, C.-Chung Chang, and C.-Jen Lin, A Practical Guide to Support Vector Classification, vol. 1, no. 1, pp. 1-16, 2010.
- [17] S. Joseph, Feature Reduction using Principal Component Analysis for Effective Anomaly – Based Intrusion Detection on NSL-KDD, *International journal of engineering science and technology*, vol. 2, no. 6, pp. 1790-1799, 2010.
- [18] M. Kaushal, A. Singh, and B. Singh, Adaptive Thresholding for Edge Detection in Gray Scale Images, vol. 2, no. 6, pp. 2077-2082, 2010.
- [19] K. Kira and L.A. Rendell, The feature selection problem: Traditional methods and a new algorithm, Proceedings of the National Conference on Artificial Intelligence, pp. 19-129, 1992.
- [20] S. Lazebnik, C. Schmid, and J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, vol. 2, pp. 2169-2178.
- [21] K.L. Murdock, SketchUp and SketchUp Pro Bible, Wiley Publishing Inc., Indianapolis, 2009.
- [22] A. Oliva and A. Torralba, Building the gist of a scene: the role of global image features in recognition, *Progress in brain research*, vol. 155, pp. 23-36, Jan. 2006.
- [23] A. Oliva, W. Hospital, and L. Ave, Modeling the Shape of the Scene : A Holistic Representation of the Spatial Envelope , vol. 42, no. 3, pp. 145-175, 2001.
- [24] E. Pekalska, D. D. Ridder, D. M. J. Tax, and S. Verzakov, A Matlab Toolbox for Pattern Recognition, PRTools version, pp. 1-61, 2007.

- [25] S. V. Rajashekararadhya and E. Engineering, Hand written numeral / mixed numerals recognition of south-Indian scripts : the zone based- feature extraction method, *Theoretical and Applied Information Technology*, 2009.
- [26] C. Rother, A. Blake, and V. Kolmogorov, GrabCut — Interactive Foreground Extraction using Iterated Graph Cuts, *ACM Transactions on Graphics SIGGRAPH'04*, 2004.
- [27] J. Shlens, A Tutorial on Principal Component Analysis, Systems Neurobiology Laboratory University of California at San Diego, 2005.
- [28] J. Song and M.R. Lyu, A Hough transform based line recognition method utilizing both parameter space and image, *Pattern recognition*, vol. 38, no. 4, pp. 539-552, 2005.
- [29] V. Vezhnevets and V. Konouchine, GrowCut - Interactive Multi-Label N-D Image Segmentation By Cellular Automata, *Proc. of Graphicon*, pp150-156, 2005.