B.N.M. Institute of Technology

An Autonomous Institution under VTU

Approved by AICTE, Accredited as Grade A Institution by NAAC till 31.12.2026.
All Eligible UG branches – CSE, ECE, EEE, ISE & Mech. Engg. Accredited by NBA till 30.06.2025
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru- 560070, INDIA
Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in, www.bnmit.org

Vidyayāmruthamashnuthe

Report on Internship

CLASSIFICATION OF SHIPS USING INVERSE SYNTHETIC APERTURE RADAR (ISAR) IMAGES

Submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

Shaurya 1BG21EC083

Under the guidance of

Dr . Dyana A

Scientist 'E'

LRDE,DRDO

Department of Electronics & Communication Engineering

2022-23

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of my internship project on Deep Learning at **Electronics and Radar Development Establishment (LRDE), Defence Research and Development Organization (DRDO)**. This project has been an incredible learning experience, and I am immensely thankful for the support and guidance provided to me throughout this journey.

I would like to extend my deepest appreciation to **Dr. Dyana A**, Scientist 'E', Electronics and Radar Development Establishment, my project supervisor, for their unwavering support, valuable insights, and continuous encouragement. Their expertise and mentorship have been instrumental in shaping my understanding of deep learning concepts and techniques. I am truly grateful for the opportunity to work under their guidance.

Finally, I would like to seize this moment to express my gratitude to my **family members** and **friends** who offered unwavering support throughout the entirety of the project.

ABSTRACT

Ship detection from Inverse Synthetic Aperture Radar (ISAR) images plays a crucial role in maritime surveillance and defence systems. However, traditional methods face challenges when dealing with complex ship shapes and difficult imaging conditions. To address these limitations, our study proposes a novel approach that combines deep learning, computer vision, YoloV4, faster RCNN and fuzzy logic techniques for ship detection in ISAR images. By leveraging the capabilities of these advanced technologies, our project aims to develop a robust and dependable system capable of accurately identifying and detecting ships. The integration of deep learning algorithms allows for the extraction of intricate features from ISAR images, while computer vision techniques enhance the understanding of ship patterns and structures. Additionally, fuzzy logic is employed to handle uncertainty and imprecision in ship detection. Through the synergistic combination of these methods, we strive to advance the state-of-the-art in ship detection and provide a valuable solution for maritime surveillance applications.

To achieve our objectives, we have outlined a comprehensive course of action. Firstly, we will generate ISAR images using MATLAB from STL files(3D model) to create a dataset for training and evaluation purposes. Subsequently, we will utilize object detection methods such as YOLOv4, RCNN, and Faster RCNN to develop a ship feature detector capable of accurately identifying ship features in the ISAR images. Finally, we will employ a Fuzzy Logic Inference System in MATLAB or Python to recognize the ship model based on the detected features.

In this study, we present a novel methodology for ship detection from Inverse Synthetic Aperture Radar (ISAR) images. Our approach demonstrates significant improvements in both accuracy and robustness compared to traditional methods, highlighting the effectiveness of incorporating deep learning techniques. The results obtained show the potential of our approach in advancing maritime surveillance and defence systems. By leveraging the power of deep learning, this research contributes to the continuous development and enhancement of ship detection capabilities in ISAR images.

TABLE OF CONTENTS

LIST OF FIGURES

ACRONYMS

ISAR	Inverse Synthetic Aperture Radar
ATR	Automatic Target Recognition
CNN	Convolutional Neural Network
ANSYS	Analysis System
DNN	Deep Neural Network
DRDO	Defence Research and Development Organisation
FIS	Fuzzy Inference System
GUI	Graphical User Interface
LRDE	Electronics and Radar Development Establishment
MF	Membership Function
MIoU	Mean Intersection over Union
NMS	Non-Maximum Suppression
ONNX	Open Neural Network Exchange
Open-CV	Open-Source Computer Vision Library
RCNN	Region-based Convolutional Neural Networks
ReLU	Rectified Linear Unit
ResNet	Residual Network
RoI	Region of Interest
RPN	Region Proposal Network
SGDM	Stochastic Gradient Descent with Momentum
VGG	Visual Geometry Group
YOLOv4	You Only Look Once version 4

1 INTRODUCTION

This chapter serves as the project introduction, providing a thorough picture of the study objectives, background knowledge, motivation for the project, and a brief overview of the report's outline. The primary objective of this project is to create a ship classification and detection system that is both efficient and precise. With a focus on using Inverse Synthetic Aperture Radar (ISAR) images, this system will be built using a combination of deep learning, computer vision, and fuzzy logic techniques. By harnessing the potential of these advanced technologies, the project aims to develop a robust and reliable system for accurately identifying and detecting ships.

1.1 Modern Developments

Ship classification and detection systems have advanced significantly in recent years because of the incorporation of cutting-edge technology such as deep learning, computer vision, and fuzzy logic approaches. The field has developed more accurate and efficient ship identification and detection capabilities by employing these advanced techniques.

Deep learning has revolutionized ship classification by enabling systems to extract intricate features from ISAR images for accurate identification. Transfer learning plays a crucial role in this process. By leveraging pre-trained models, such as ResNet or VGGNet, which are trained on large-scale datasets for general object recognition tasks, the system can benefit from the learned knowledge and then fine-tune the models specifically for ship classification. This approach significantly reduces the need for extensive training on limited ship datasets, allowing for faster and more accurate classification results.

In the detection of ships, computer vision techniques have been crucial. RCNN (Region-based Convolutional Neural Networks), Faster R-CNN (Region-based Convolutional Neural Networks), and YOLOv4 (You Only Look Once) are notable approaches. Because of its effectiveness and precision, YOLOv4 is a real-time object identification system that excels in spotting ships in complicated environments. To recognize ships within images and offer accurate bounding box localization, RCNN and Faster R-CNN use region suggestions. The performance of ship detection is improved by these computer vision techniques, allowing for more efficient maritime monitoring and security.

Fuzzy logic techniques provide a valuable framework for ship classification and detection systems to handle uncertainty and imprecise data. In ship classification tasks, fuzzy logic can account for ambiguous features and provide a more nuanced decision-making process.

By allowing the representation of vague or incomplete information, fuzzy logic enhances the system's ability to handle noise, variability, and imprecision in ship classification and detection.

The integration of deep learning with transfer learning, computer vision techniques (such as YOLOv4, RCNN, Faster R-CNN), and fuzzy logic has significantly advanced ship classification and detection systems. Transfer learning enables the extraction of intricate features from ISAR images, enhancing accuracy and reducing training time. Computer vision techniques provide efficient and accurate ship detection capabilities. Additionally, fuzzy logic techniques enable robust handling of uncertainty and imprecise data. These modern developments have improved maritime surveillance, security, and operational efficiency.

1.2 Problem Statement

Apart from target detection and tracking, Automatic Target Recognition (ATR) is gaining importance in contemporary radar systems. For high-resolution images of marine targets, Inverse Synthetic Aperture Radar (ISAR) technology is used. It is necessary for sea targets (ships) for naval purposes to be classified. Using image processing methods, ISAR images are pre-processed. Convolutional neural networks can be used to classify the ship. Then, using machine learning algorithms, object features (such as ship features) are retrieved and categorized. Object detection algorithms like YOLOv4, RCNN, and Faster-RCNN for ship feature detection.

1.3 Objective

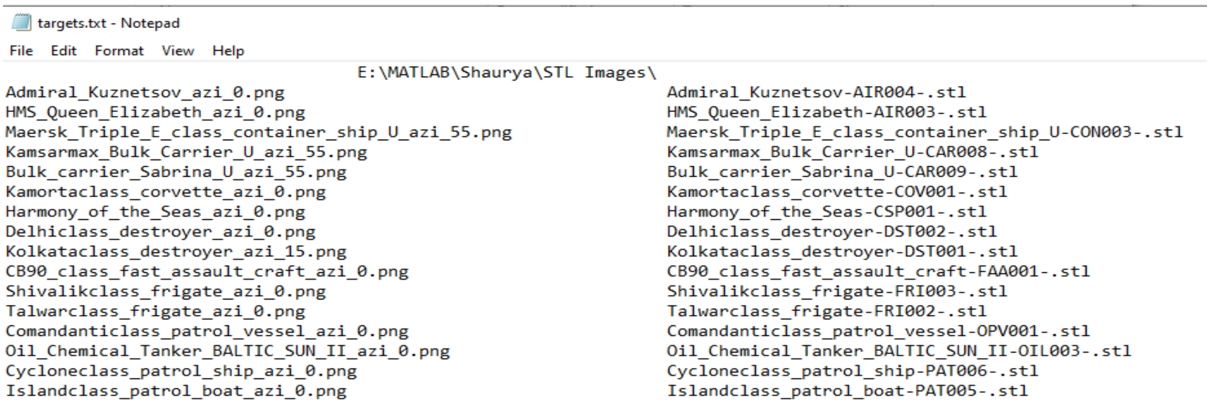
- Database creation with the help of Electromagnetic Simulation tool and Image Labeler Tool.
- Design and develop Convolution Neural Network based architecture for classification of ships.
- Design and develop Object Detector for detecting ship features using algorithms like YOLOv4, RCNN and Faster RCNN.
- Develop ship recognition model based on the detected ship features using Fuzzy Logic.
- Validate the performance of the system using the database created.

1.4 Strategy

The course of action for tackling these objectives is to generate ISAR images with MATLAB and ANSYS, train a modified VGG-16 model for real-time ship classification, make a ship feature detector using object detection methods like YOLOv4, RCNN, and Faster RCNN, and recognize the ship model using a Fuzzy Logic Inference System in MATLAB or Python.

• TARGET text file

This file contains the SHIP'S name that are being classified and STL file along with their path. There are 10 categories and 16 ships in total.



targets.txt - Notepad

File Edit Format View Help

```
                                E:\MATLAB\Shaurya\STL Images\
Admiral_Kuznetsov_azi_0.png                               Admiral_Kuznetsov-AIR004-.stl
HMS_Queen_Elizabeth_azi_0.png                             HMS_Queen_Elizabeth-AIR003-.stl
Maersk_Triple_E_class_container_ship_U_azi_55.png         Maersk_Triple_E_class_container_ship_U-CON003-.stl
Kamsarmax_Bulk_Carrier_U_azi_55.png                       Kamsarmax_Bulk_Carrier_U-CAR008-.stl
Bulk_carrier_Sabrina_U_azi_55.png                         Bulk_carrier_Sabrina_U-CAR009-.stl
Kamortaclass_corvette_azi_0.png                           Kamortaclass_corvette-COV001-.stl
Harmony_of_the_Seas_azi_0.png                             Harmony_of_the_Seas-CSP001-.stl
Delhiclass_destroyer_azi_0.png                            Delhiclass_destroyer-DST002-.stl
Kolkataclass_destroyer_azi_15.png                         Kolkataclass_destroyer-DST001-.stl
CB90_class_fast_assault_craft_azi_0.png                   CB90_class_fast_assault_craft-FAA001-.stl
Shivalikclass_frigate_azi_0.png                           Shivalikclass_frigate-FRI003-.stl
Talwarclass_frigate_azi_0.png                             Talwarclass_frigate-FRI002-.stl
Comandanticlass_patrol_vessel_azi_0.png                   Comandanticlass_patrol_vessel-OPV001-.stl
Oil_Chemical_Tanker_BALTIC_SUN_II_azi_0.png               Oil_Chemical_Tanker_BALTIC_SUN_II-OIL003-.stl
Cycloneclass_patrol_ship_azi_0.png                        Cycloneclass_patrol_ship-PAT006-.stl
Islandclass_patrol_boat_azi_0.png                         Islandclass_patrol_boat-PAT005-.stl
```

Figure 1.1: Targets text file

• ISAR image database creation

To generate ISAR images, we use STL file which have 3-dimensional model of all the ships. First, we will create .bin file from .stl file then we have to set the yaw, pitch and roll for the ship movement in x ,y and z axis(0.01,1,0.01) . Change the Az(Azimuth) value to 0,5,10,15,40,45 in sq_angle . i value will iterate till 300 to make 1 bin file and after 27 bin file iteration will stop .Similarly we have to create .bin file for all the ships along with all azimuth value and save it in a new folder Testbinfiles.

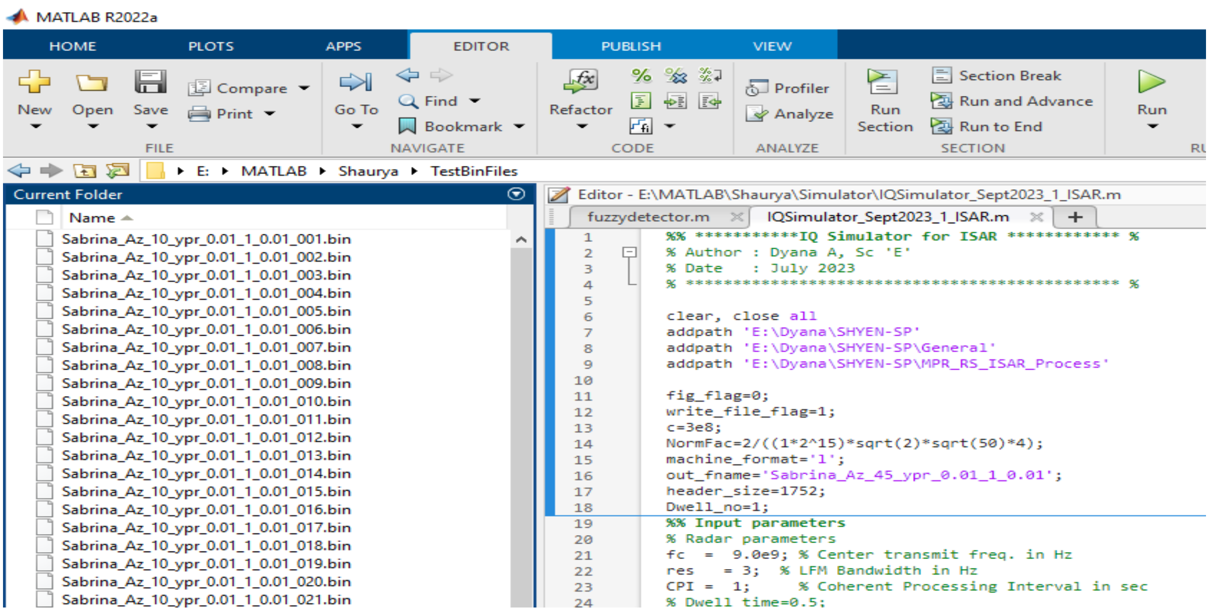


Figure 1.2: .bin file(IQSimulator)

- **Design and Training of Neural Network**

A customized variant of the convolution architecture is created for extracting features and performing classification tasks. The architecture is adjusted to reduce the number of layers and is trained in MATLAB.

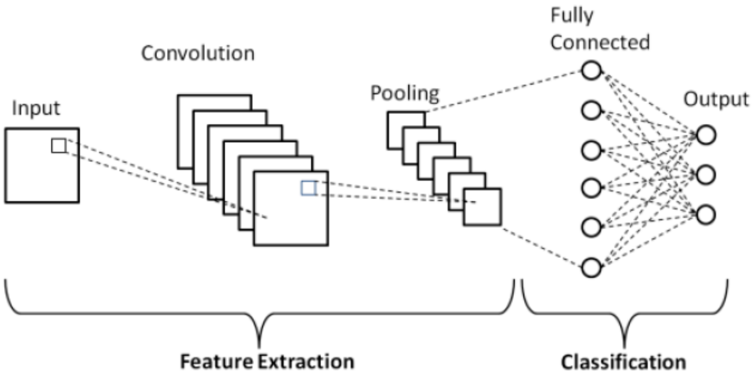


Figure 1.1: General CNN Architecture

- **SP_main(creation of isar images)**

Here we are creating ISAR images from .bin file .First we will create ISAR_SHIPS_DB6 in which we have 10 categories and 16 ships in total .

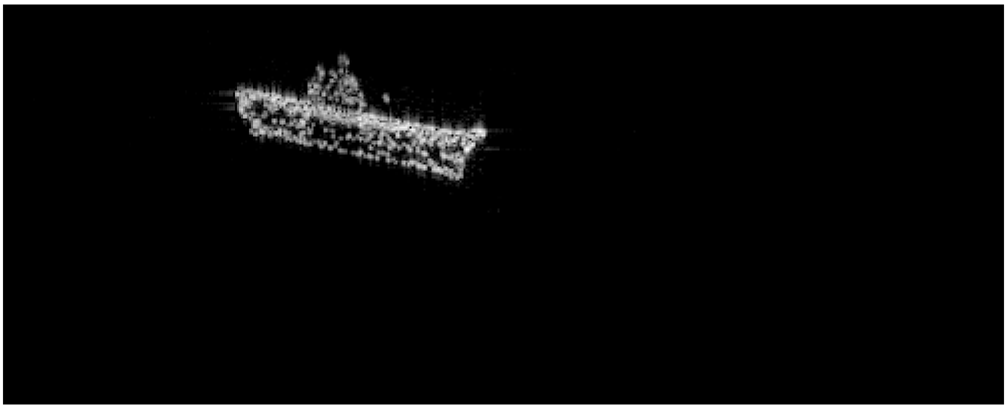


Figure 1.4: ISAR image of aircraft carrier admral

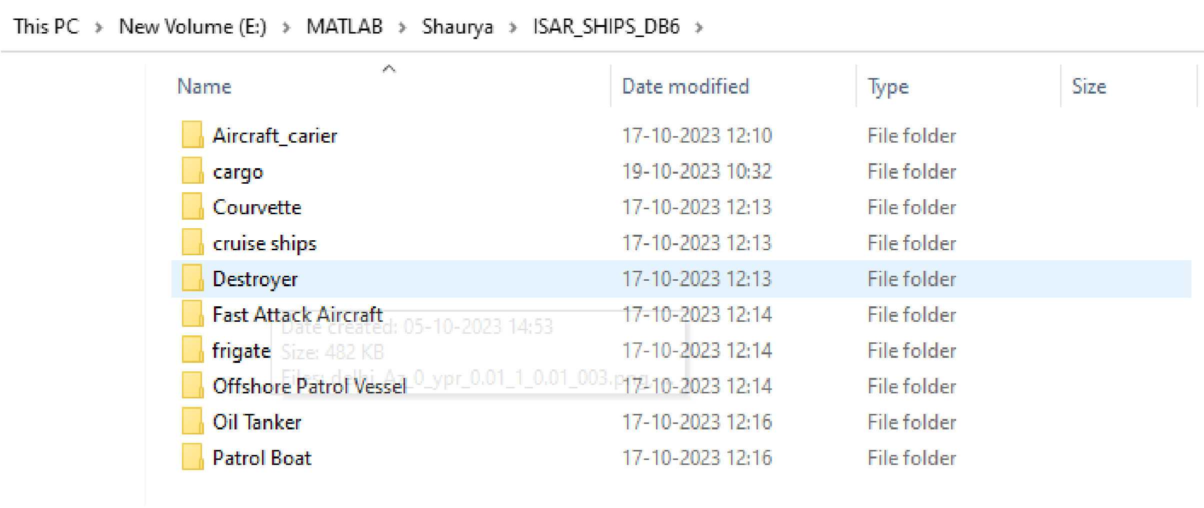


Figure 1.5: ISAR folder

- **Design and Training of Object Detector**

The Object Detector based on YOLOv4 had its neural network composed of backbone, neck and head in which variant of VGG-16 was used for backbone. Additionally, another Object detector was made using Faster-RCNN composing feature extraction layer, box regression layer, region proposal network (RPN) classifier, region of Interest (RoI) Layer and Classification Layer. For Feature Extraction Layer the same architecture as backbone of YOLOv4 was used. The detector is used for detecting required ship features for recognizing the ship model.

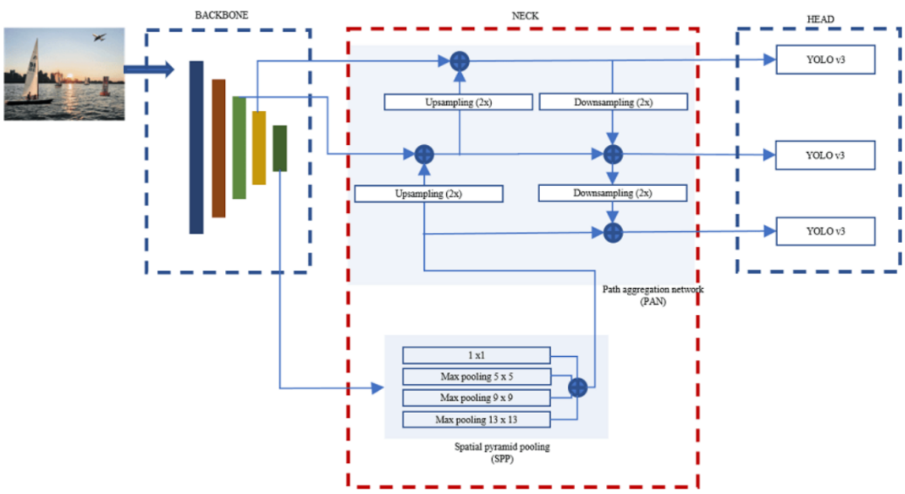


Figure 1.6: General YOLOv4 Architecture

Create a different folder for Training (Eg.ISAR_SHIPS_Training) copy image from ISAR_SHIPS_DB6 *we are removing some ships because they look similar and they have same super structure. Net file will be created .

Training on single CPU.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Loss	Mini-batch Accuracy	Mini-batch RMSE	RPN Mini-batch Accuracy	RPN Mini-batch RMSE	Base Learning Rate
1	1	00:00:01	1.5787	32.26%	0.30	47.66%	0.51	1.0000e-06
3	315	00:02:20	1.5065	98.10%	0.30	54.69%	0.60	1.0000e-06
5	630	00:04:39	1.5162	99.55%	0.26	60.16%	0.72	1.0000e-06
7	945	00:07:00	1.3719	98.89%	0.25	65.62%	0.52	1.0000e-06
7	1043	00:08:21	1.4696	99.40%	0.30	61.72%	0.70	1.0000e-06

Training finished: Max epochs completed.
Detector training complete.

Figure 1.7: Training

After training we will get a accuracy-loss graph in which we have 10 epoch :

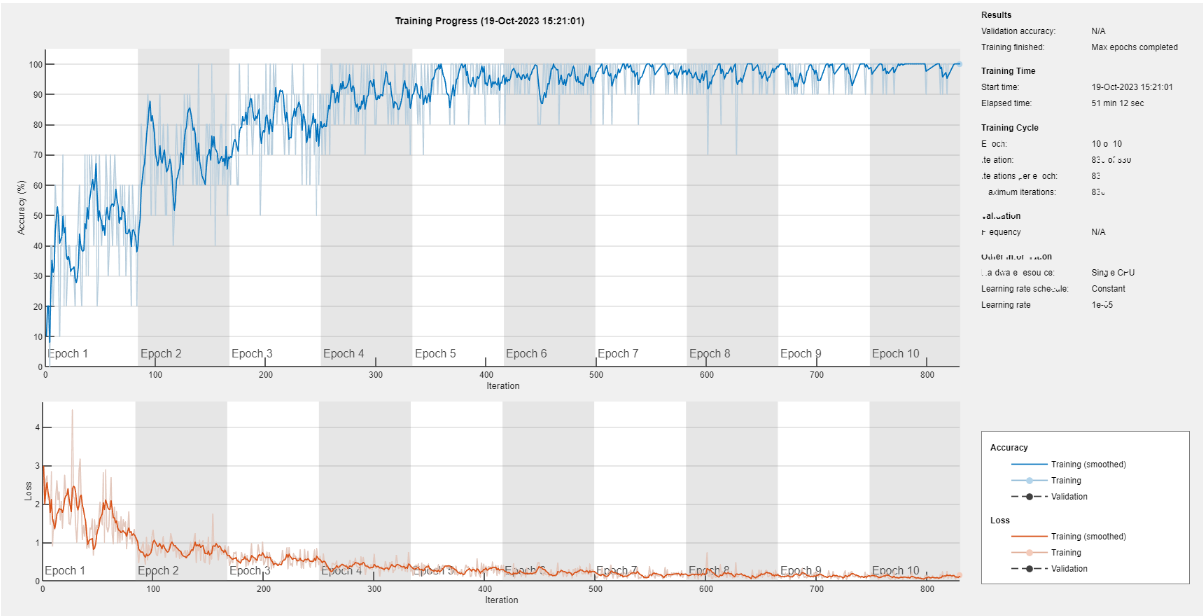


Figure 1.8: Accuracy Loss Graph

- **Developing Ship Recognition Model**

Developing a ship recognition model involves utilizing ship features and employing a fuzzy inference system. Ship features, such as number of super structures and location of super structure are extracted from the input data. These features serve as inputs to a fuzzy inference system, which is designed to handle uncertainties and imprecise information. The fuzzy inference system utilizes a set of linguistic variables, membership functions, and a rule base to perform inference and make decisions based on the extracted ship features.

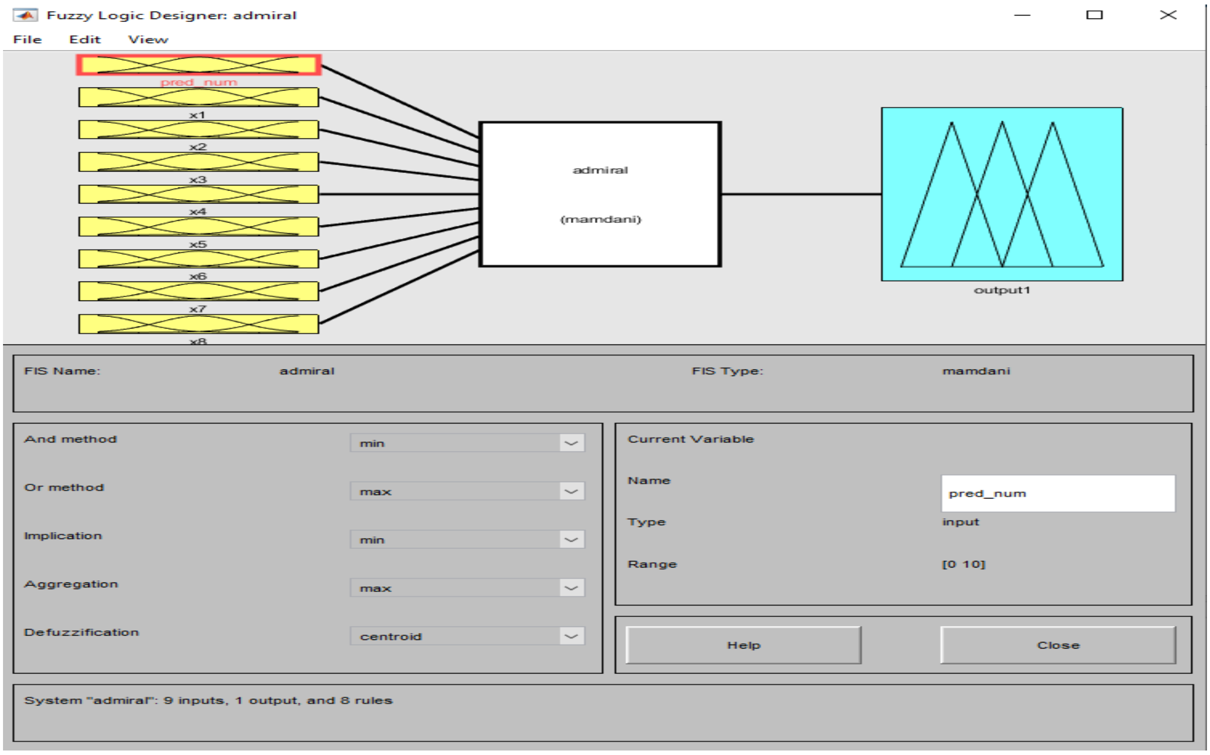


Figure 1.9: Fis pred_num, rules, output

1.5 Summary

The organizational structure of the project is briefly summarized in this chapter. From the introduction through the conclusion, the report's seven chapters are outlined, with each chapter's goal and content highlighted. Readers may follow the logical flow of information because of the framework, which ensures clear and thorough coverage of the project.

2 OVERVIEW

This chapter gives an overview of the use of deep learning, computer vision, and fuzzy logic techniques for classifying and detecting ships using ISAR images. It emphasizes the value of neural networks as well as the applicability of deep learning, computer vision, and fuzzy logic techniques to the project. An input ISAR image of a ship is first classified using a deep learning technique known as Convolutional Neural Networks (CNNs), then ship features are extracted using object detection algorithms like YOLOv4 and Faster-RCNN, and finally, with the extracted ship features, the ship model is detected using Fuzzy Logic.

2.1 Ship Classification

Classifying ships from Inverse Synthetic Aperture Radar (ISAR) images is pivotal for enhancing maritime surveillance and security. Convolutional Neural Networks (CNNs), a prominent deep learning architecture in image recognition, play a pivotal role in this endeavor. Renowned models such as VGG16 and ResNet are considered for crafting the neural architecture, leveraging transfer learning to expedite training and enhance performance on generated ISAR images. The study delves into the impact of Adagrad, SGDM (Stochastic Gradient Descent with Momentum), and Adam optimizers on convergence speed and accuracy.

Furthermore, the exploration extends to regularization techniques like L1 and L2 regularization to curb overfitting and enhance the model's generalization. The efficacy of the ship classification model is gauged through diverse performance metrics, including F1 score, accuracy, precision, recall, and a confusion matrix. These metrics offer a nuanced understanding of the model's proficiency in discerning various ship classes.

To facilitate the development and evaluation process, popular deep learning libraries such as Keras, PyTorch, and TensorFlow are harnessed. These libraries provide a rich repertoire of pre-implemented CNN layers, optimizers, and evaluation metrics, streamlining the efficient development and evaluation of the ship classification model.

Ship Feature Detection

The application of object detection algorithms, specifically YOLOv4 and Faster R-CNN, was employed to recognize ship models using classified Inverse Synthetic Aperture Radar (ISAR) images. The primary goal is to extract key ship features, including the number of Super Structures and their respective locations, identified within bounding boxes.

YOLOv4, denoting "You Only Look Once version 4," stands out as a cutting-edge object detection algorithm, celebrated for its exceptional speed and accuracy. Comprising three integral components—the backbone, neck, and head—this algorithm is structured to optimize the extraction of high-level features. The backbone initiates the process with initial feature extraction layers, capturing the essence of input images. The neck, comprising feature fusion and upsample layers, works to enhance spatial resolution and amalgamate multi-scale features, thereby refining object localization. Finally, the head undertakes the critical tasks of object classification and bounding box regression.

In contrast, Faster R-CNN, an abbreviation for "Faster Region-based Convolutional Neural Networks," is a widely acknowledged object detection algorithm renowned for its accuracy and precise object localization. Its backbone network, often rooted in architectures like VGG16 or ResNet, undertakes feature extraction from input images. The Region Proposal Network (RPN) takes charge of generating region proposals by predicting scores and bounding box coordinates. These proposed regions traverse the RoI (Region of Interest) pooling layer, extracting fixed-size feature maps. The final steps involve further processing by fully connected layers, culminating in object classification and bounding box regression.

YOLOv4 and Faster R-CNN represent two prominent paradigms in object detection, each boasting unique strengths and characteristics. YOLOv4, with its real-time capabilities, excels in applications demanding swift processing, employing a single-stage approach and advanced backbone networks. On the other hand, Faster R-CNN, a two-stage algorithm, is distinguished by its precision in object localization and resilience in detecting smaller objects, utilizing RPNs and CNN-based feature extraction. The choice between the two hinges on specific requirements, whether prioritizing real-time detection and overall performance or placing a premium on accuracy and meticulous object localization.

2.2 Ship Identification

The detected ship features, such as number of super structures and their location, are utilized to identify ship models. By applying fuzzy logic, the system can handle uncertainty and imprecise information in ship model recognition. The integration of object detection algorithms and fuzzy logic inference provides a comprehensive approach to accurately recognize ship models from ISAR images.

Faster R-CNN and YOLOv4 are employed for ship feature detection in ISAR images. These object detection algorithms effectively identify and localize ship features, namely superstructures, facilitating subsequent ship model recognition.

A fuzzy logic inference system is integrated into the ship model recognition process to handle uncertainty and imprecision. The system incorporates linguistic variables, membership functions, and a rule base to interpret the ship features and make informed decisions for ship model identification.

The integration of Faster R-CNN and YOLOv4 for ship feature detection, followed by a fuzzy logic inference system for ship model recognition, presents a comprehensive framework. This approach enables accurate identification and classification of ship models from ISAR images. The proposed methodology contributes to enhancing maritime surveillance, security, and operational efficiency.

2.3 Summary

This chapter provides an overview on several machine learning concepts like Convolutional Neural Networks (CNN), Object Detection Algorithms (YOLOv4 and Faster-RCNN) and Fuzzy Logic used across the project for classifying and identifying the ship from the input ISAR images.

3 SHIP CLASSIFICATON

An overview of the architectural choices made, and design factors considered for the DNN system utilized for the ship classification system is provided in this chapter. This chapter acts as a guide for the implementation phase, guaranteeing the efficient design and development of the ship image classification system that utilizes deep neural networks and ISAR images. It serves as a blueprint, providing a framework for constructing the system and ensuring its successful realization.

3.1 Neural Network Architecture

The architecture utilized for ship classification involves a tailored version of the well-known VGG-16 network. VGG-16, a highly acclaimed deep convolutional neural network, is esteemed for its exceptional performance in tasks related to image classification. Our customized adaptation involves precise modifications to the layer quantity and filter configurations, strategically fine-tuned to enhance the network's efficacy in ship classification. These alterations are geared towards attaining a harmonious blend of high accuracy and computational efficiency during the processing of ship images .

The neural network architecture is mainly composed of these layers:

1. Convolutional Layer 1:

• Number of filters: 32

• Activation function: ReLU

• Pooling: Max pooling

2. Convolutional Layer 2:

• Number of filters: 32

• Activation function: ReLU

• Pooling: Max pooling

3. Convolutional Layer 3:

• Number of filters: 64

• Activation function: ReLU

• Pooling: Max pooling

4. Flatten Layer: Converts the multidimensional output of the previous layer into a one-dimensional vector.

5. Fully Connected Layers:

• **Layer 1**: 64 neurons with ReLU activation

• **Dropout:** 10% dropout rate to prevent over-fitting.

• **Layer 2:** 7 neurons (corresponding to the 7 ship classes) with softmax activation

for classification.

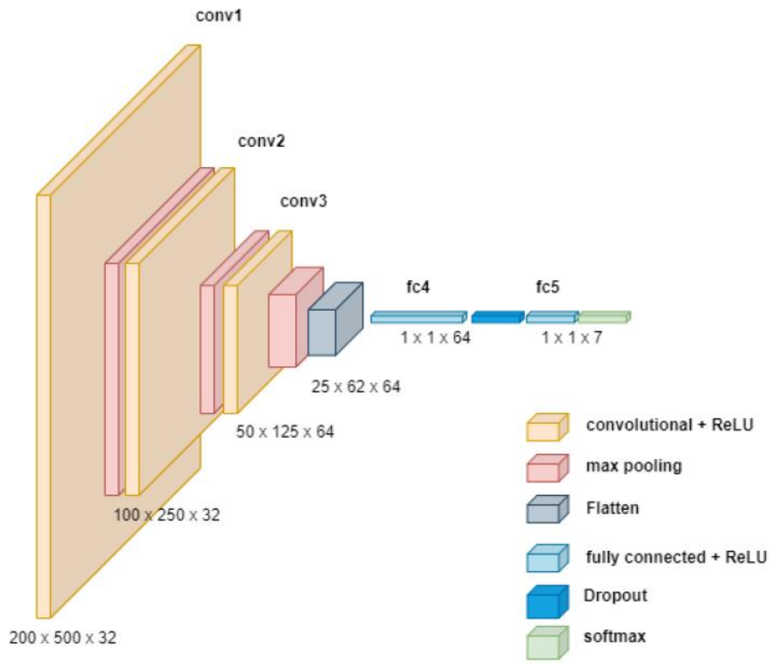


Figure 3.1: Classification Neural Network

The ship classes are namely, Aircraft Carrier, Container, Frigate, Fishing, Oil Tanker, Passenger and Patrol.

The selected network architecture is designed to effectively capture significant features from ISAR images, enabling accurate ship classification predictions. Max pooling is utilized to down sample the images, reducing spatial dimensions while extracting crucial information. This approach manages computational complexity while preserving important details for classification purposes.

With a total of 6,377,383 trainable parameters, the network undergoes training using TensorFlow and Keras in Python. The training dataset comprises 2400 ISAR ship images across seven distinct classes, aiming to optimize the network's weights for precise classification of unseen images.

To enhance performance, the Adagrad optimizer is employed to mitigate overfitting when dealing with large datasets. Data augmentation techniques, including horizontal and vertical image flipping, are applied to simulate real-world scenarios and diversify the training samples. Additionally, image normalization ensures compatibility with other system modules, promoting consistent and reliable results.

Throughout the training process, evaluation metrics such as recall, F1 score, and precision are computed to assess the network's performance. These metrics provide valuable insights into the effectiveness of the trained neural network for accurate ship classification, offering a quantitative measure of its capabilities.

3.2 Classification Module

The Classification Module incorporates a C++ code that imports a pre-trained neural network model in ONNX format. The code processes input images from the radar controller, normalizes them, and passes them through the neural network. The output consists of confidence scores for 7 classes, with the highest score indicating the predicted ship class. Implemented with Open-CV DNN libraries, the module enables real-time ship classification from ISAR images, ensuring efficient execution and suitability for resource-constrained environments.

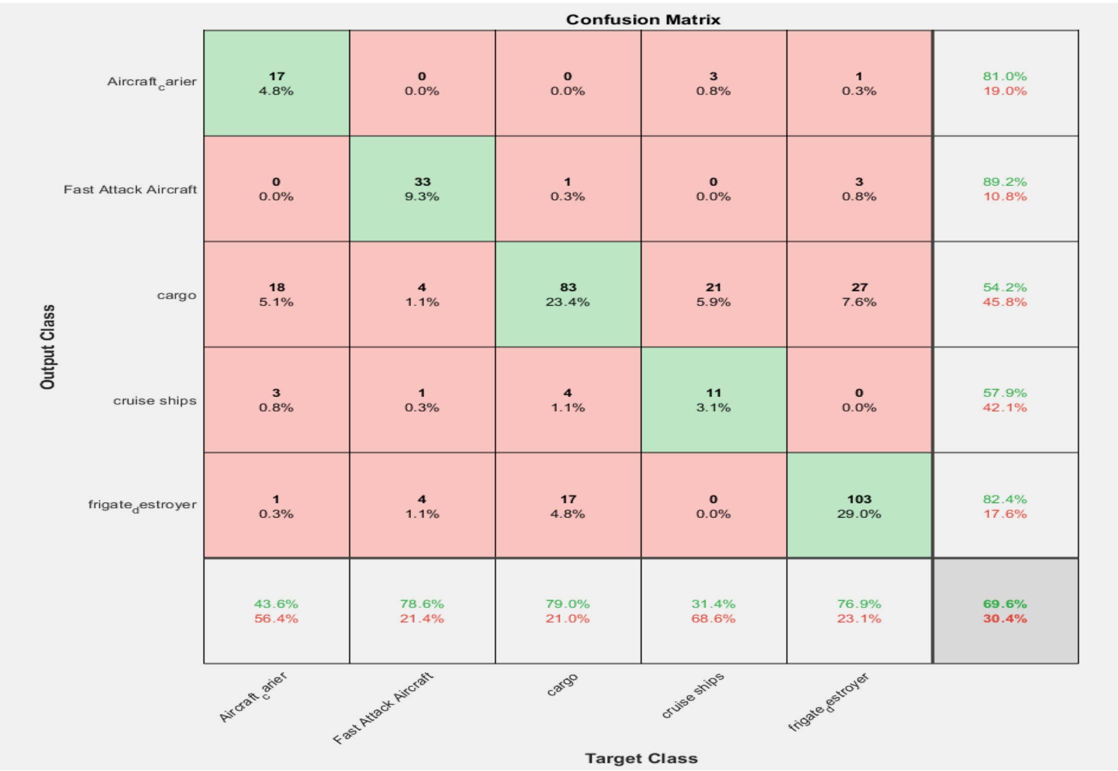


Figure 3.2: Confusion Matrix for Ship Classification System

3.3 Summary

This chapter provides an overview of the architectural choices and design factors for the ship classification system utilizing deep neural networks and ISAR images. The selected neural network architecture is a modified version of VGG-16, optimized for ship classification while maintaining computational efficiency. The network includes convolutional layers, max pooling, and fully connected layers. The system employs TensorFlow and Keras for training, with data augmentation and the Adagrad optimizer to enhance performance. The Classification Module incorporates a C++ code with Open-CV DNN libraries for real-time ship classification from ISAR images. The chapter serves as a blueprint for efficient system design and development.

4 SHIP FEATURE DETECTION

This chapter focuses on finding out different feature aspects of the input ISAR image ship after its classification. Features like the number of superstructures and their location are used to recognizing the ship model. These features can be visualized from the ISAR images. Hence, Object Detection Algorithms were incorporated to detect these features (superstructures) from the ISAR images.

4.1 Database for Object Detection

The main objective is to extract ship features, particularly the number of Super Structures and their locations, from the detected bounding boxes. The focus is on identifying different feature aspects of the ship after its classification, enabling the recognition of specific ship models. The integration of object detection algorithms facilitates the detection of these features, specifically the superstructures, from the ISAR images. Therefore, Object detection algorithms, namely YOLOv4 and Faster R-CNN, for ship model recognition using classified Inverse Synthetic Aperture Radar (ISAR) images.

Creating a database for object detection of superstructures from ISAR images, specifically for the implementation of YOLOv4 and Faster R-CNN, involves utilizing the MATLAB Image Labeler app. The MATLAB Image Labeler app provides a user-friendly interface for annotating images and generating ground truth data.

In the first step, the ISAR images are loaded into the MATLAB Image Labeler app, which allows for precise annotation of the superstructures. The app provides various annotation tools such as bounding boxes to mark the locations of the superstructures within the images. The user can interactively label each superstructure with accurate boundaries. This process ensures the creation of a high-quality and reliable database.

Once the annotations are complete, the MATLAB Image Labeler app facilitates the export of the labeled data in a format compatible with YOLOv4 and Faster R-CNN algorithms. The exported data includes the image files and their corresponding annotations, allowing for seamless integration with the training pipeline of the object detection algorithms. By utilizing the MATLAB Image Labeler app, the process of creating a well-annotated database becomes efficient and convenient, laying the foundation for training robust YOLOv4 and Faster R-CNN models for superstructure detection in ISAR images.

Ships like Admiral Carrier , HMS queen, Harmony of the sea, Delhi class, Kolkata class etc. we have created a different data base for all the ships in a folder named labelled images

Eg.-As there were limited accurate models available for container ships like Bulk Carrier Sabrina, Kamsarmax Bulk Carrier, and Maersk Triple E-Class Container, a dedicated database was created for this specific ship class using the MATLAB Image Labeller App. A total of 149 ISAR images were carefully annotated using the app to label each superstructure visible within the image for container.

The database included a comprehensive set of 149 images, each with annotations marking the location and boundaries of every superstructure present in the ISAR image. The MATLAB Image Labeler App provided a user-friendly interface for precise annotation, ensuring accurate labeling of the superstructures.

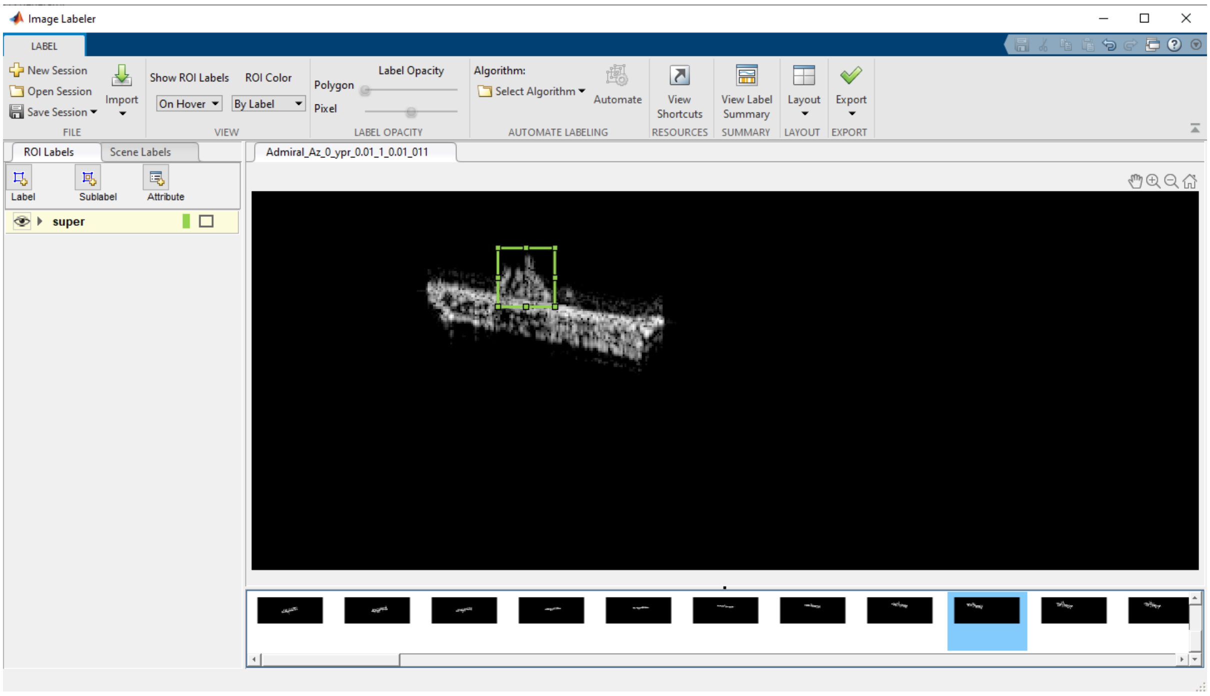


Figure 4.1: Image Labeller used for labelling Admiral

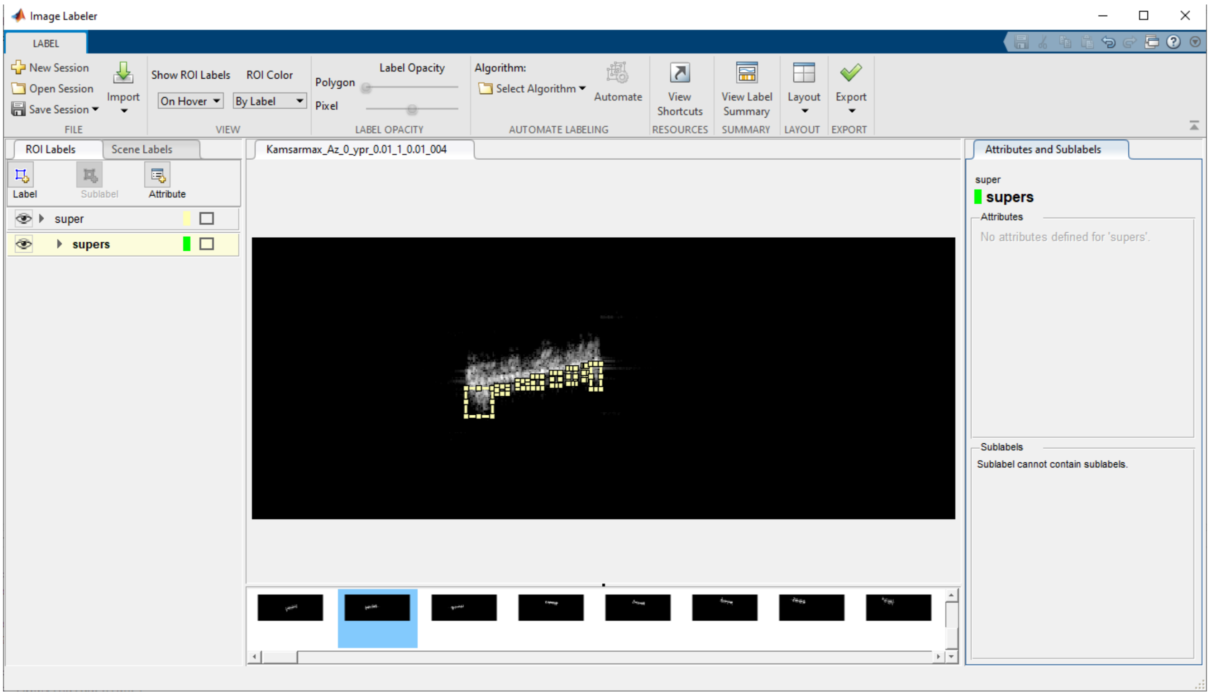


Figure 4.2: Image Labeller used for labelling Kamsarmax

4.2 Ship Feature Detector

Given the availability of accurate ship computer-aided design models for object detection, for ships like Admiral, HMS, Bulk Carrier Sabrina, Kamsarmax Bulk Carrier, Maersk Triple E-Class Container, Kamorta class, harmony, Delhi class , Kolkata class, CB90, Shivalik, Talwar class, etc. a specific ship feature detector was developed. This detector focused on ships belonging to the Container Class , Aircraft Carrier, Corvette, Cruise, Destroyer, Fast Attack Aircraft category. To accomplish this, both YOLOv4 and Faster R-CNN object detection algorithms were employed to create separate models.

The objective was to compare the results obtained from the two algorithms and determine which model would be the preferred choice. By conducting a thorough evaluation and comparison, the most suitable algorithm for accurately detecting ship features of container ships could be identified.

Speed: YOLOv4 is known for its real-time object detection capabilities. It processes the entire image in a single forward pass, making it faster than Faster R-CNN, which involves multiple stages and region proposals. In this scenario where real-time processing is crucial, YOLOv4 might be a better choice.

Single Stage Detection: YOLOv4 is a single-stage object detection algorithm, meaning it directly predicts bounding boxes and class probabilities in one pass. This can simplify the training and deployment processes and may lead to faster convergence.

Accuracy: While the speed of YOLOv4 is a significant advantage, YOLOv4 also aims to maintain competitive accuracy. It includes various enhancements over previous versions, incorporating features to improve detection performance.

Flexibility: YOLOv4 can handle detection of multiple object classes simultaneously in a given image, making it suitable for scenarios where there are multiple types of objects, such as ships along with other objects of interest.

The ship feature detector generates bounding box annotations, which are used to determine the number and location of superstructures on a ship. The location is expressed as a percentage relative to the beginning of the ship. These results are then given as input into ship detection models that utilize fuzzy logic to identify the ship model that best matches the ISAR image.

4.2.1 YOLOv4 Detector

YOLOv4 is a renowned object detection algorithm known for its speed and accuracy. It comprises three essential components: the backbone, neck, and head. The backbone extracts high-level features from input images, while the neck refines the spatial resolution and combines multi-scale features for precise object localization. The head handles object classification and bounding box regression. YOLOv4 incorporates advanced techniques like anchor-based prediction, L1 and L2 regularization for generalization, and optimization techniques such as Adagrad, SGDM, and Adam.

In the first step, YOLOv4 performs initial feature extraction using the deep convolutional neural network (CNN) mentioned above. This network extracts high-level features from the input image, capturing important visual representations.

Next, the algorithm performs object detection by applying a series of convolutional layers to the extracted features. These layers are responsible for predicting bounding boxes, object classes, and confidence scores. YOLOv4 predicts a fixed number of bounding boxes per grid cell, along with the associated class probabilities. The algorithm employs anchor boxes to handle objects of various sizes and aspect ratios effectively. The size of these anchor boxes is initially estimated using a function in the MATLAB Computer Vision Toolbox. Six anchor boxes were estimated for Sabrina and the sizes of each are[61 23;24 22;21 12;22 9;17 10;1210].

The network is then trained with the 149 image database. The precision of predicted bounding box is determined by confidence score and Mean Intersection over Union (mIoU). The confidence score is calculated by multiplying the objectness score and the class probability. The objectness score represents how likely a bounding box contains an object, while the class probability indicates the likelihood of the object belonging to a specific class.

Mean Intersection over Union is a metric used to evaluate the quality of object detection models. It measures the degree of overlap between the predicted bounding boxes and the ground truth bounding boxes. The mIoU score calculates the intersection over union ratio for each pair of predicted and ground truth bounding boxes, and then takes the average of these ratios.

The threshold input argument in the detect function of MATLAB is used to control the minimum confidence score required for an object detection to be considered valid. This threshold value determines the level of certainty needed to accept a detected object. Objects with confidence scores above the threshold are considered as valid detections, while those below the threshold are filtered out as false positives or uncertain detections. The threshold for the model based on YOLOv4 was calculated by subtracting the maximum confidence score for the specific input ISAR image with a constant value.

The optimal constant value was determined through a process of testing various potential values and evaluating the associated factors, such as errors in the predicted number of boxes and annotations of the predicted boxes. By analysing these factors, the value that resulted in the minimum errors was identified as the most suitable constant value. Through rigorous testing the optimal value came to 0.0068.

In YOLOv4, Non-Maximum Suppression (NMS) is a post-processing step that uses an overlap threshold of 0.05 to determine the degree of overlap between bounding boxes for suppression. The overlap threshold, measured as Intersection over Union (IoU), specifies the minimum intersection area divided by the union area for two bounding boxes to be considered overlapping.

```
 1   'input'       Image Input   200×500×1 images
 2   'conv_1'      Convolution   32 3×3×1 convolutions with stride [1  1] and padding 'same'
 3   'relu_1'      ReLU          ReLU
 4   'maxpool_1'   Max Pooling   2×2 max pooling with stride [2  2] and padding [0  0  0  0]
 5   'conv_2'      Convolution   32 3×3×32 convolutions with stride [1  1] and padding 'same'
 6   'relu_2'      ReLU          ReLU
 7   'conv_3'      Convolution   64 3×3×32 convolutions with stride [1  1] and padding 'same'
 8   'relu_3'      ReLU          ReLU
 9   'conv_7'      Convolution   36 3×3×64 convolutions with stride [1  1] and padding 'same'
10   'relu_7'      ReLU          ReLU
```

Figure 4.3: YOLOv4 Neural Architecture

4.2.2 Faster-RCNN Detector

Faster R-CNN, also referred to as "Faster Region-based Convolutional Neural Networks," is a widely used object detection algorithm renowned for its precise localization and high accuracy. The backbone network conducts feature extraction from input images. The Region Proposal Network (RPN) generates region proposals by predicting bounding box coordinates and scores. These proposals are then passed through the ROI (Region of Interest) pooling layer, extracting feature maps of fixed size. Subsequently, fully connected layers process these features, leading to object classification and bounding box regression.

In the first step, the backbone network, extracts high-level features from the input image. This feature extraction process enables the network to capture rich representations of the image, capturing important visual information.

Next, the Region Proposal Network (RPN) operates on the feature maps produced by the backbone network. The RPN generates region proposals by predicting both the objectness scores (to determine if an object is present) and the corresponding bounding box coordinates. These region proposals are potential regions of interest that are likely to contain objects. Thirteen anchor boxes were estimated and the sizes of each are [61 23;24 21;23 12;23 9;18 10;13 10;16 22;19 26;23 32;28 30;33 40;99 136;119 163].

Once the region proposals are generated, they are passed through the ROI pooling layer. This layer extracts fixed-size feature maps from each region proposal. These feature maps are then fed into fully connected layers, where they undergo classification and bounding box regression. The classification phase assigns a class label to each region proposal, indicating the type of object it represents. The bounding box regression phase adjusts the predicted bounding box coordinates to align more accurately with the ground truth bounding boxes.

The training process involves optimizing the network's parameters using labelled data with ground truth annotations. This includes adjusting the RPN and the classification and regression layers to minimize the discrepancy between predicted and ground truth values.

The threshold value, which is a parameter that is used to determine the minimum confidence score required for a detection to be considered valid, was kept at 0.5. In Non-Maximum Suppression (NMS), which is a post-processing step in Faster R-CNN, the overlap threshold is a parameter used to determine the degree of overlap between bounding boxes that would trigger suppression and it was kept at 0.05. The overlap threshold, often represented as an IoU (Intersection over Union) value, defines the minimum intersection area divided by the union area required for two bounding boxes to be considered overlapping.

```
 1   'input'                           Image Input                 200×500×1 images with 'zerocenter' normalization
 2   'conv1'                           Convolution                 64 7×7×1 convolutions with stride [2  2] and padding [3  3  3  3]
 3   'bn_conv1'                        Batch Normalization         Batch normalization with 64 channels
 4   'relu_conv1'                      ReLU                        ReLU
 5   'maxpool1'                        Max Pooling                 3×3 max pooling with stride [2  2] and padding [1  1  1  1]
 6   'conv2_1|conv1'                   Convolution                 64 64×64×64 convolutions with stride [1  1] and padding 'same'
 7   'conv2_1|bn1'                     Batch Normalization         Batch normalization with 64 channels
 8   'conv2_1|relu1'                   ReLU                        ReLU
 9   'maxpool_5'                       Max Pooling                 2×2 max pooling with stride [2  2] and padding [0  0  0  0]
10   'conv'                            Convolution                 32 3×3×64 convolutions with stride [1  1] and padding [1  1  1  1]
11   'relu'                            ReLU                        ReLU
12   'conv1x1'                         Convolution                 26 1×1×32 convolutions with stride [1  1] and padding [0  0  0  0]
13   'conv1x1 box regression (RPN)'    Convolution                 52 1×1×32 convolutions with stride [1  1] and padding [0  0  0  0]
14   'proposalLayer'                   Region Proposal             region proposal with 13 anchor boxes
15   'roi pooling layer'               ROI Max Pooling             ROI Max Pooling with pooled output size [15 15]
16   'fc_1'                            Fully Connected             64 fully connected layer
17   'relu_3_1'                        ReLU                        ReLU
18   'fc_2'                            Fully Connected             2 fully connected layer
19   'softmax'                         Softmax                     softmax
20   'fc_reg'                          Fully Connected             4 fully connected layer
21   'classoutput'                     Classification Output       crossentropyex with classes 'super' and 'Background'
22   'boxRegression'                   Box Regression Output       smooth-l1 loss
23   'smooth-l1'                       Box Regression Output       smooth-l1 loss
24   'rpn_softmax'                     RPN Softmax                 rpn softmax
25   'RPN Classification Output'       RPN Classification Output   cross-entropy loss with 'object' and 'background' classes
```

Figure 4.4: Architecture of Faster-RCNN

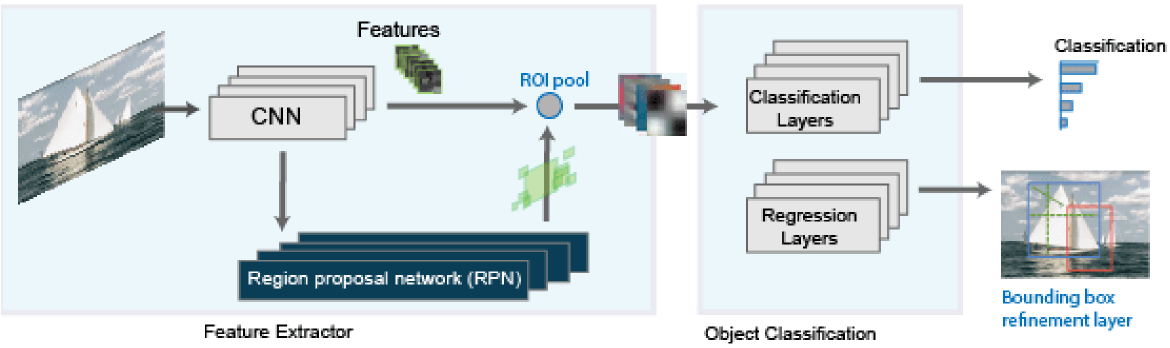
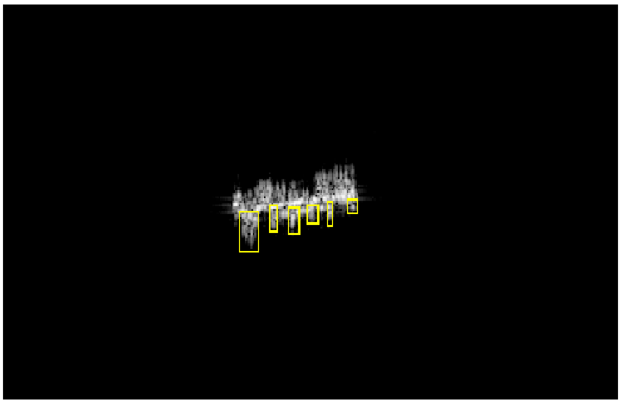


Figure 4.5: General Architecture of Faster-RCNN

4.3 Summary

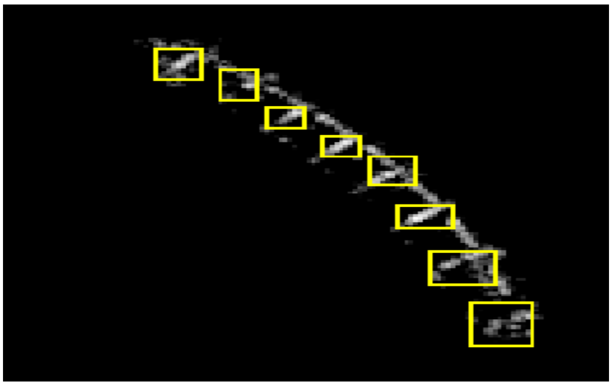
In this chapter, the focus is on extracting different ship features, such as the number of superstructures and their locations, from ISAR images after ship classification. Object detection algorithms, YOLOv4 and Faster R-CNN, are incorporated to detect these features from the ISAR images. The MATLAB Image Labeler app is utilized to create a database for object detection, where the ISAR images are annotated with precise bounding boxes for the superstructures. The database is then used to train YOLOv4 and Faster R-CNN models for ship feature detection. The threshold and overlap values are carefully chosen to ensure accurate detection and suppression of overlapping bounding boxes. Overall, this chapter establishes the framework for accurately recognizing and localizing ship features using object detection algorithms and ISAR images.

(a). Bulk Carrier Sabrina (6 ground truth super structure)



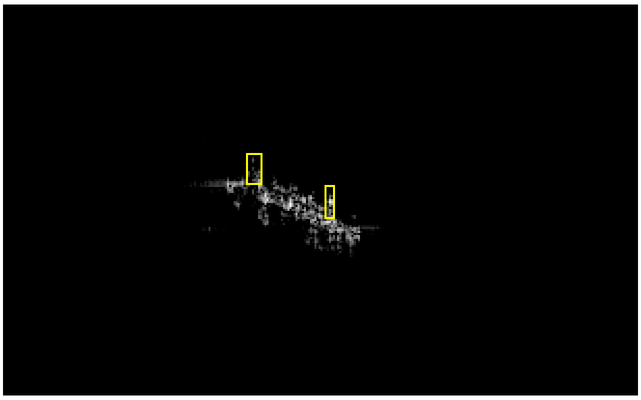
Bulk Carrier Sabrina
Yaw Rate = 0.01 rad/sec
Pitch Rate = 1 rad/sec
Roll Rate = 0.01 rad/sec
Azimuth Angle = 5 degree

(b). Kamsarmax Bulk Carrier (8 ground truth super structure)



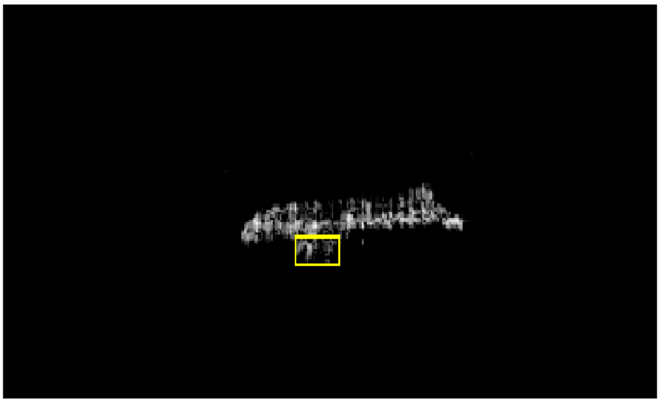
Kamsarmax Bulk Carrier
Yaw Rate = 0.01 rad/sec
Pitch Rate = 1 rad/sec
Roll Rate = 0.01 rad/sec
Azimuth Angle = 5 degree

(c). Maersk Triple-E Class Container (2 ground truth super structure)



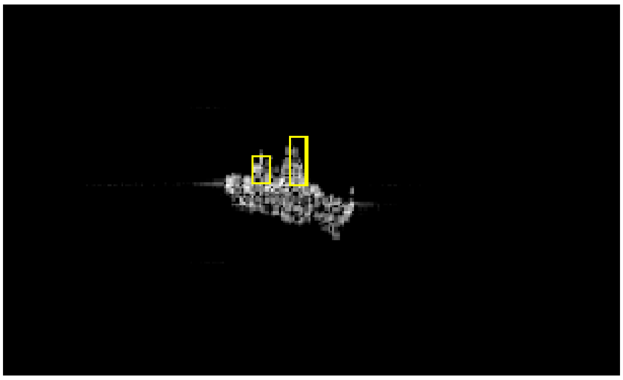
Maersk Triple-E Class Container
Yaw Rate = 0.01 rad/sec
Pitch Rate = 1 rad/sec
Roll Rate = 0.01 rad/sec
Azimuth Angle = 5 degree

(d). Admiral(Aircraft Carrier) (1 ground truth super structure)



Admiral(Aircraft Carrier)
Yaw Rate = 0.01 rad/sec
Pitch Rate = 1 rad/sec
Roll Rate = 0.01 rad/sec
Azimuth Angle = 5 degree

(d). Kolkata Class (2 ground truth super structure)



Kolkata Class
Yaw Rate = 0.01 rad/sec
Pitch Rate = 1 rad/sec
Roll Rate = 0.01 rad/sec
Azimuth Angle = 5 degree

Figure 4.6: Output of YOLOv4 Ship Feature Detector for Input ISAR images

In the realm of ship superstructure detection, the accurate assessment of various parameters plays a pivotal role in ensuring the reliability of computer vision models. The evaluation metrics encompassing the number of unchanged boxes, number of small changes in the number of boxes, and errors in calculating the number of boxes serve as critical benchmarks for model performance. The former two metrics gauge the model's ability to maintain consistency and discern subtle modifications in the detected superstructures, while the latter measures the precision of box count predictions. Additionally, the X-coordinate error, Y-coordinate error, height error, and width error of predicted boxes offer valuable insights into the localization accuracy of the detected superstructures. These metrics collectively contribute to the refinement and enhancement of ship superstructure detection algorithms, fostering advancements in maritime surveillance and safety.

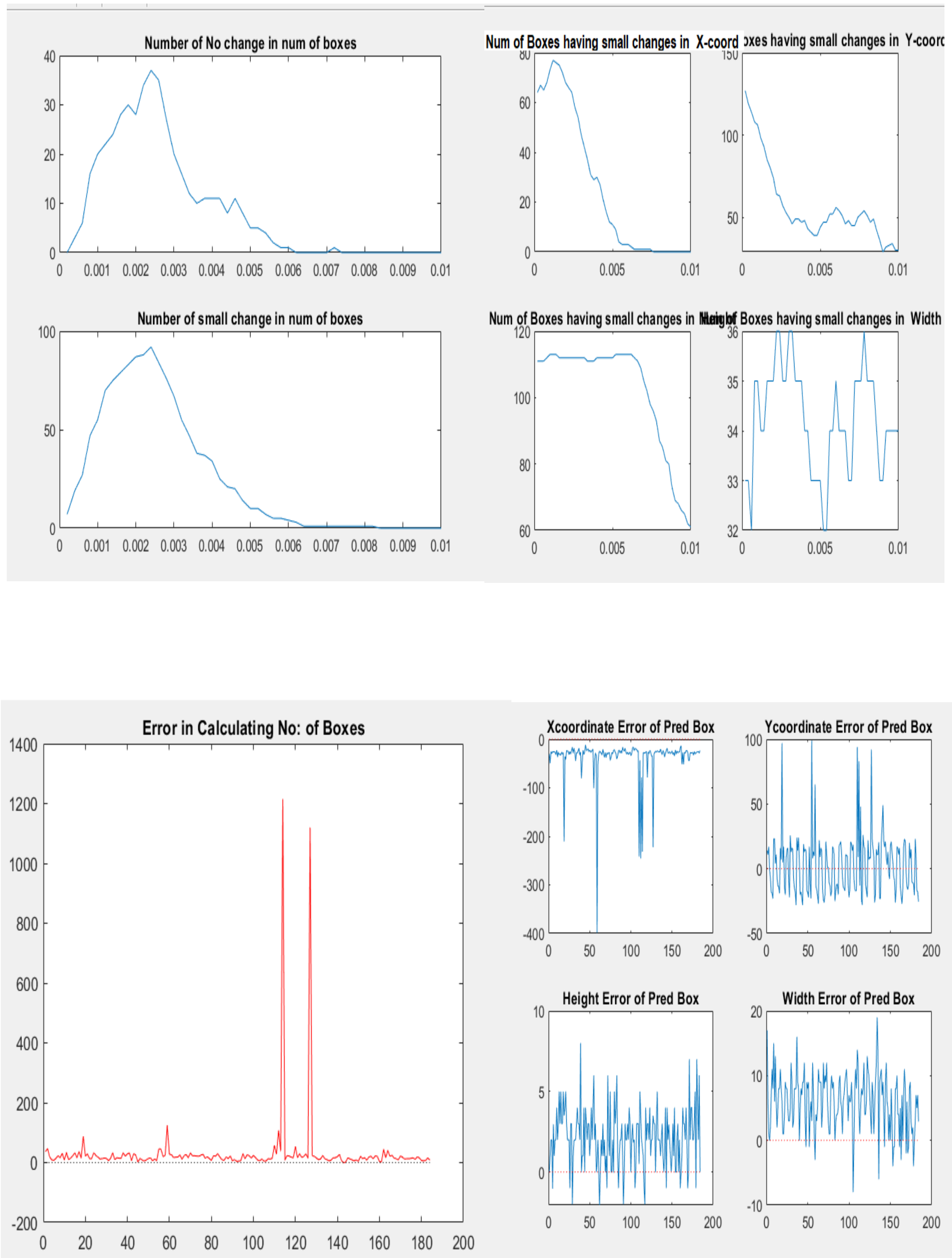


Figure 4.7:MATLAB figures for:

i) Number of No & small change in number of boxes
ii) Num of Boxes having small changes in X-Y-coordinate
iii) Error in calculating number of boxes
iv) X-Ycoordinate Error of Pred Box, Height-Width Error of Pred Box

5 SHIP IDENTIFICATION

This chapter delves into the detection of ship models using the output generated by the ship feature detector model. The primary objective is to improve recognition accuracy by incorporating Fuzzy Logic techniques. Fuzzy Logic is implemented to handle the inherent uncertainties and vagueness associated with ship model recognition.

5.1 Fuzzy Logic

Fuzzy Logic is a powerful mathematical framework that deals with uncertainty and imprecision in decision-making processes. It has found wide applications in various fields, including control systems, pattern recognition, and decision support systems. MATLAB's Fuzzy Logic Toolbox provides a user-friendly environment called Fuzzy Logic Designer, which allows users to design and simulate fuzzy logic systems easily.

In Fuzzy Logic Designer, one of the key components is membership functions. Membership functions define the degree of membership or relevance of an input or output variable to a particular fuzzy set. MATLAB provides a range of predefined membership functions such as triangular, trapezoidal, and Gaussian, or users can define custom membership functions to capture specific characteristics of the problem domain.

Another important aspect in Fuzzy Logic Designer is the definition of fuzzy rules. Fuzzy rules establish relationships between the input and output variables. These rules are typically expressed in the form of "IF-THEN" statements, where the antecedent (IF part) specifies the condition based on the input variables, and the consequent (THEN part) determines the output variables. The Fuzzy Logic Designer allows users to define and edit fuzzy rules intuitively, facilitating the development of complex fuzzy systems.

The Mamdani method, named after its creator Ebrahim Mamdani, is a popular inference method used in fuzzy logic systems. It employs fuzzy logic to derive crisp output values based on fuzzy input variables and predefined fuzzy rules. The Mamdani method combines the degrees of membership from the fuzzy sets defined by the input variables according to the fuzzy rules to obtain a final output value. This method is well-suited for applications where linguistic rules and fuzzy reasoning are required.

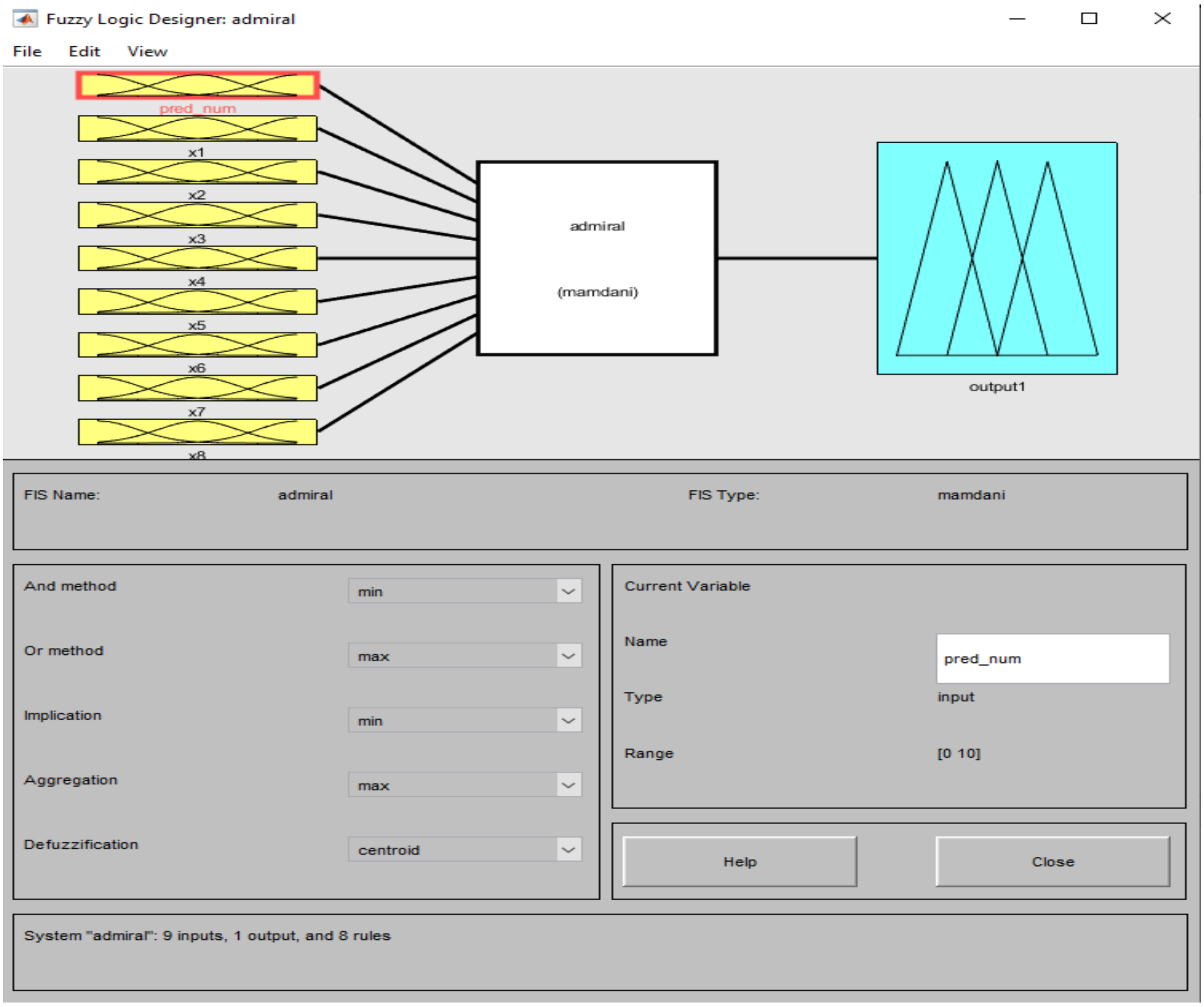


Figure 5.1: Admiral Fuzzy Inference System

5.2 Fuzzy Inference System

The output from the ship feature detector, which is the form of number and location of superstructures on a ship (the location is expressed as a percentage relative to the beginning of the ship), is fed into a fuzzy inference system (FIS) made using MATLAB's Fuzzy Logic Designer app.

There are total of nine FIS models in which three FIS models are for the container class ship feature detector, which represented each ship model present in the container class (bulk carrier Sabrina, Kamsarmax bulk carrier, and Maersk triple E-class container) and other are for Admiral, Harmony, Kolkata class, Delhi class, Shivalik class, Talwar class . Each FIS model was ship specific. The number of inputs varied with each FIS model. For example, for the FIS model of the bulk carrier Sabrina, there were 7 inputs, while for the Kamsarmax bulk carrier, there were 9 inputs. The input number is based on the number of predicted superstructures and the number of ground-truth superstructures.

In the following table :

1st column is the predicted number (pred_num) in which bad=1, bad1=3, good=2

2nd – 9th column is the input in which bad=2, bad1=3, good=1

10th column is the output in which Not_name=1, Maybe_name=2, Name=3

Editor - FIS_generatorshaurya.m | Variables - ruleList

ruleList

8x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	1	1	1	1	2	2	2	1	1
2	0	1	1	1	1	1	2	2	2	1	1	1
3	1	0	0	0	0	0	0	0	0	1	1	1
4	3	0	0	0	0	0	0	0	0	1	1	1
5	0	1	1	1	1	1	1	1	2	3	1	1
6	0	0	0	0	0	0	0	2	2	1	1	1
7	2	1	1	1	1	1	1	1	1	3	1	1
8	3	1	1	1	1	1	1	1	1	1	1	1

Figure 5.2: Rules of FIS generator

5.2.1 Fuzzification

One of the inputs (pred_num) is the number of predicted superstructures, which is indicated by the number of predicted boxes by the ship feature detector. The membership functions (mf) used for this input were one gaussian mf, one s-shaped mf, and one z-shaped mf. The gaussian mf represented the "good" aspect of the input and the other two "bad" and "bad1" aspects of the input, respectively. The gaussian mf was centred around the value, which corresponds to the number of ground truth superstructures. These various aspects are then used for making rules, which are the factors that determine the final output.

The rest of the inputs indicate the distance between a predicted superstructure and the closest ground truth superstructure. If the predicted number of superstructures is less than the number of ground truth superstructures, then extra superstructures are assumed to be present at a location that is much farther from any of the ground truth superstructures as a penalty measure (say at a position that is 200 percent relative to the beginning of the ship).

The membership functions for these inputs are then made based on the inputs, which are an absolute (modulus) measure of the distance. Small values, like values under 5 units, are the "good" aspect of the input, which is indicated by a z-shaped mf, and rest values are the "bad" aspect of the input and are indicated by an s-shaped mf.

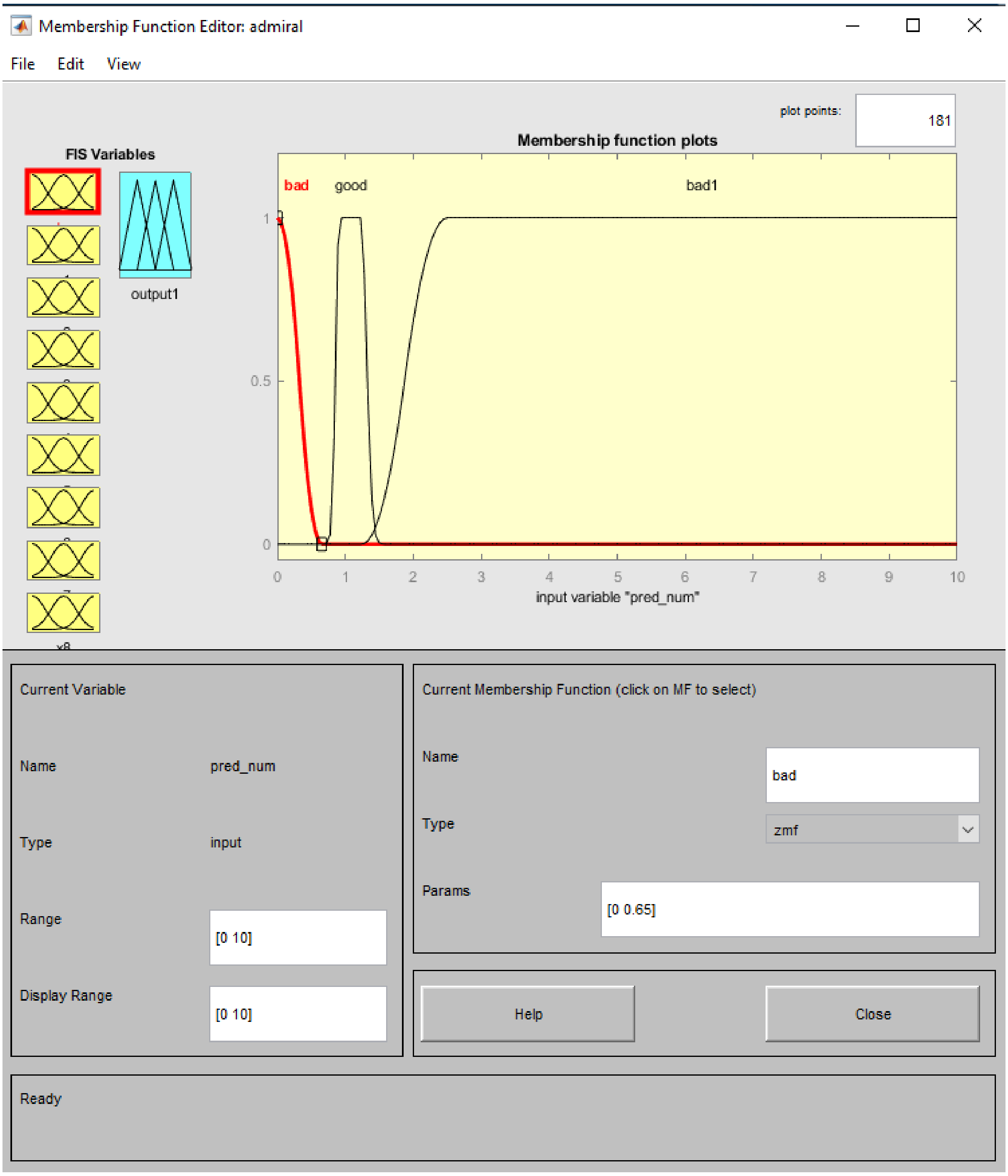


Figure 5.3: Membership Functions of pred_num of Admiral FIS

5.2.2 Rule Evaluation

The next step in making an FIS model is to specify the rules to be used for this FIS model. These rules are made by carefully inspecting the ISAR image database and the results from the ship feature detector. Each rule has statements (representing a value) that are connected by OR or AND conjunction words. OR operation is interpreted by the FIS as maximum function, whereas AND is interpreted as minimum function.

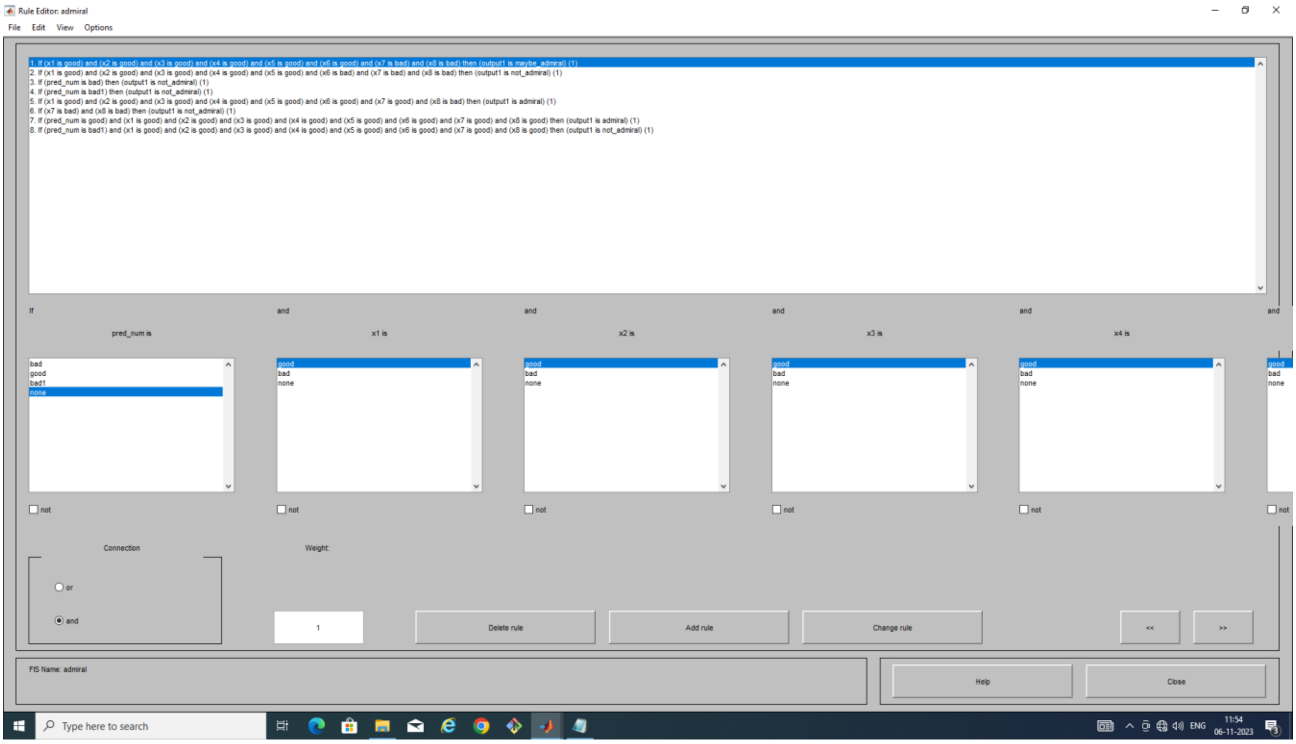


Figure 5.4: Rules Defined for Admiral FIS

5.2.3 Defuzzification

The output is designed so that the final value can be any number between 0 and 100. Lower results (say below 40) correspond to the score that show the ISAR picture ship model (represented by z-shaped mf) is not the exact model from which the FIS model is made. The possibility that the model indicated in the ISAR picture may be the particular model from which the FIS model is made (expressed by the combination of two Gaussian membership functions, or gauss2mf) corresponds to values in the middle of the range, say between 40 and 75. An increased number, such as one above 75, indicates a greater likelihood that the model is the particular model from which the FIS model is made (indicated by an s-shaped mf).

The results from the ship feature detector are passed to every FIS model present under the specific class the input ISAR image ship has been classified under. The output from every FIS model is the observed ability to recognize and detect the ship model.

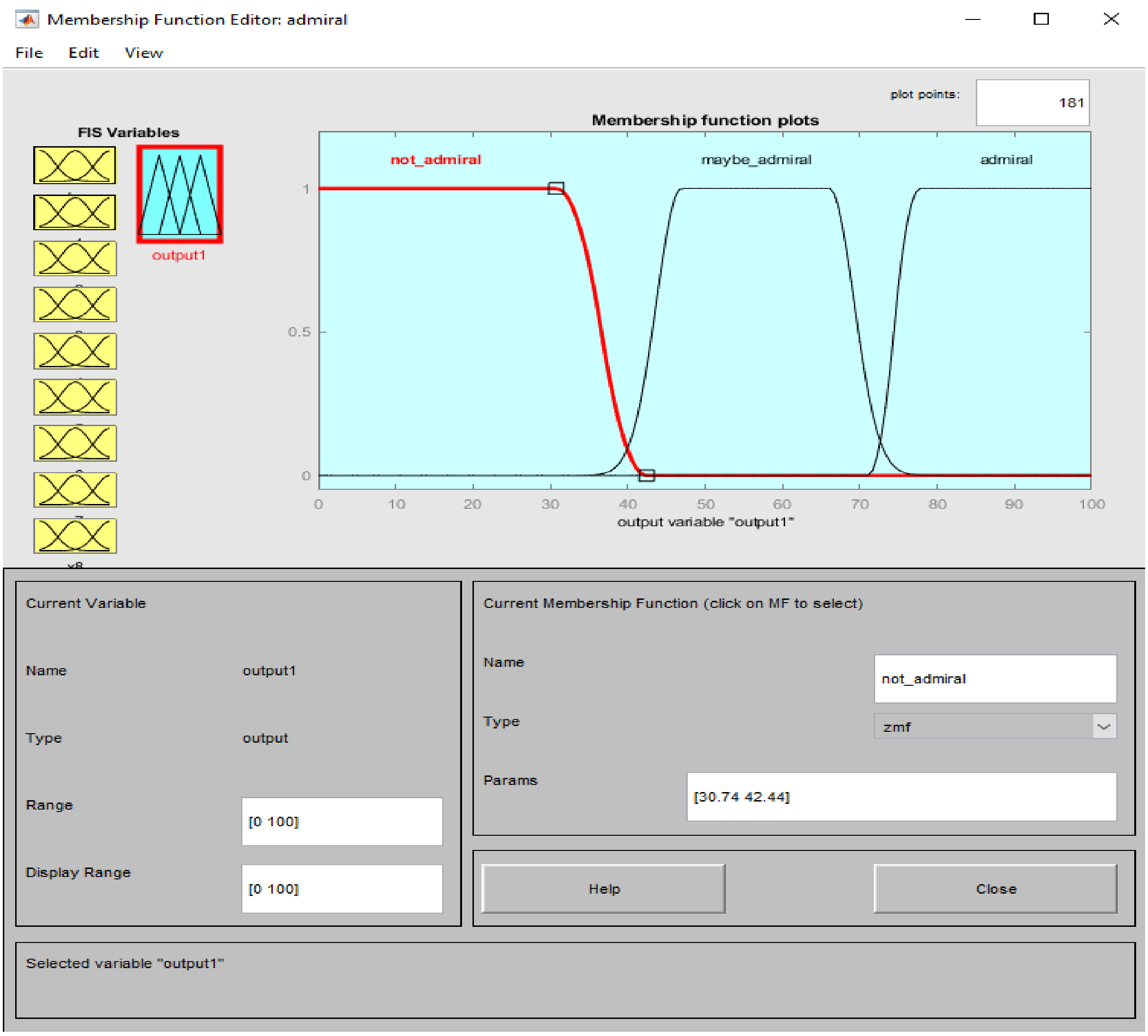


Figure 5.5: Output Membership Functions of Admiral

5.3 Summary

This chapter focuses on ship model detection using the output generated by a ship feature detector model. The primary objective is to improve recognition accuracy by incorporating Fuzzy Logic techniques. Fuzzy Logic is employed to handle uncertainties and vagueness associated with ship model recognition, providing a robust framework for identification. MATLAB's Fuzzy Logic Toolbox and Fuzzy Logic Designer app are utilized for designing and simulating the Fuzzy Inference System (FIS).

The chapter also explains the fuzzy inference process, including fuzzification, rule evaluation, and defuzzification. The ship feature detector's output, which includes the number and location of superstructures on a ship, is passed to the FIS models specific to each ship model within the container class. The FIS models evaluate the inputs, apply fuzzy rules, and produce output values indicating the likelihood of a match between the ship model and the input data.

6 CONCLUSION

This project focuses on the development of an advanced ship detection system specifically designed for ISAR (Inverse Synthetic Aperture Radar) images. By leveraging the capabilities of deep learning, computer vision, and fuzzy logic techniques, this system achieves remarkable results in ship detection and classification. The integration of these innovative technologies lays a solid foundation for future advancements in ship detection and classification technology, offering potential improvements in data-set size, real-world performance, and computational efficiency. The system's successful application to ISAR images contributes significantly to the field of ship detection and propels advancements in radar-based object recognition.

6.1 Limitations of the Project

1.Computational Challenges:

- Enhancing the neural architecture for ship classification and feature detection proved to be computationally demanding.

- Training and validating the neural network required substantial resources, potentially limiting real-time performance in resource-constrained environments.

- Mitigating this challenge involves optimizing the network architecture and exploring hardware acceleration techniques.

2. Data Availability Constraints:

- The project relied on an ISAR image dataset from the Indian Navy and Indian Air Force.

- To enhance system generalization and robustness, it is recommended to include a broader range of ship classes and diverse imaging scenarios.

- Expanding the dataset is crucial for improving the system's capabilities.

3. Performance Navigation Challenges:

- The system's effectiveness may be compromised under demanding imaging conditions, such as low image quality or variations in ship orientation.

- Improving performance in these scenarios requires exploring additional pre-processing techniques or alternative network architectures.

4. Addressing Hardware Limitations:

- The system's implementation was tailored to a specific hardware setup, limiting compatibility with different platforms.

- To enhance performance and scalability, it is essential to ensure compatibility with various hardware configurations and optimize the system for specialized platforms.

5. Limited Real-Time Evaluation:

- The system's performance evaluation was confined to offline analysis using pre-processed ISAR images.

- Comprehensive understanding of practical applicability and effectiveness requires assessing the system in real-time scenarios.

- Real-time evaluation is crucial for insights into the system's performance under dynamic conditions and accurate assessments of real-world potential.

6. Future Enhancements:

- Future developments should focus on dataset extension, addressing computational complexity, improving performance in challenging environments, real-time performance validation, and optimizing hardware compatibility.

6.2 Potential Improvements

Improving the classification of ships using Inverse Synthetic Aperture Radar (ISAR) images involves enhancing various aspects of the system. Here are some potential improvements to consider:

Augmented Dataset:

Expand the dataset by including a more diverse set of ship classes, spanning different sizes, shapes, and configurations. A larger and more representative dataset can improve the model's ability to generalize to a broader range of scenarios.

Data Augmentation:

Apply data augmentation techniques to artificially increase the size of the training dataset. Techniques such as rotation, flipping, and scaling can help the model become more robust and invariant to variations in ship orientation and imaging conditions.

Transfer Learning:

Leverage pre-trained models on large datasets and fine-tune them on your ISAR ship classification task. Transfer learning can expedite the training process and potentially improve the model's performance, especially when the available dataset is limited.

Advanced Neural Architectures:

Experiment with state-of-the-art neural network architectures. Consider architectures designed specifically for image classification tasks, such as EfficientNet, or explore newer versions of YOLO or other object detection models that may have been released since your initial implementation.

Ensemble Learning:

Combine the predictions of multiple models using ensemble techniques. Ensemble learning, such as bagging or boosting, can enhance classification accuracy by leveraging the strengths of different models and mitigating their individual weaknesses.

Attention Mechanisms:

Integrate attention mechanisms within the neural network architecture. Attention mechanisms can help the model focus on relevant regions of the ISAR images, improving its ability to capture fine-grained details crucial for accurate ship classification.

Robust Pre-processing Techniques:

Implement and experiment with advanced pre-processing techniques to enhance image quality. Address challenges such as noise reduction, resolution enhancement, and contrast adjustment to improve the overall quality of ISAR images before feeding them into the model.

Adaptive Learning Rate Scheduling:

Implement adaptive learning rate scheduling strategies during training to ensure efficient convergence and prevent overshooting. Techniques such as learning rate annealing or cyclical learning rates can help fine-tune the model more effectively.

Quantization and Compression:

Explore model quantization and compression techniques to reduce the computational and memory requirements of the model, making it more suitable for deployment in resource-constrained environments.

6.2.1 Evaluation Metrics

Here, four main evaluation metrics are used to assess the system's performance:

1. Accuracy:

• Formula: Accuracy =

$$\frac{\text{Number of correctly detected samples}}{\text{Total number of samples}}$$

• Accuracy measures the overall correctness of the detection results. It indicates the proportion of correctly detected samples out of the total number of samples.

2. Precision:

• Formula: Precision =

$$\frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

• Precision focuses on the positive predictions made by the system. It measures the proportion of correctly predicted positive samples out of all the samples predicted as positive.

3. Recall (Sensitivity or True Positive Rate):

• Formula: Recall =

$$\frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

• Recall measures the proportion of correctly predicted positive samples out of all the actual positive samples. It indicates the system's ability to correctly identify positive instances.

4. F1 Score:

• Formula: F1 Score =

$$\frac{2\times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

• The F1 score combines precision and recall into a single metric. It provides a balanced evaluation of the model's performance, considering both the positive prediction accuracy and the ability to correctly identify positive instances. By calculating these evaluation metrics based on the detection results obtained from the ship detection system, it becomes possible to accurately assess the system's performance and compare different models or approaches values for these metrics for the system.

6.2.2 Experimental Dataset

The Experimental Dataset comprising 159 gray-scale images in PNG format with 16-bit depth per pixel was used. These images are divided into 3 ship models, namely Bulk Carrier Sabrina (59 images), Kamsarmax Bulk Carrier (48 images) and Maersk Triple-E Class Container (52 images). Each image has dimensions of 200 x 500 pixels and is stored as a gray-scale image with a 16-bit depth. The data set was generated through the processing of signal files obtained from ANSYS using MATLAB.

6.2.3 Performance Analysis

This section evaluates the ship detection system using ISAR images.

Evaluation Metrics for the ship detection system:

Accuracy	0.9268
Precision	0.9268
Recall	0.9268
F1 score	0.9268

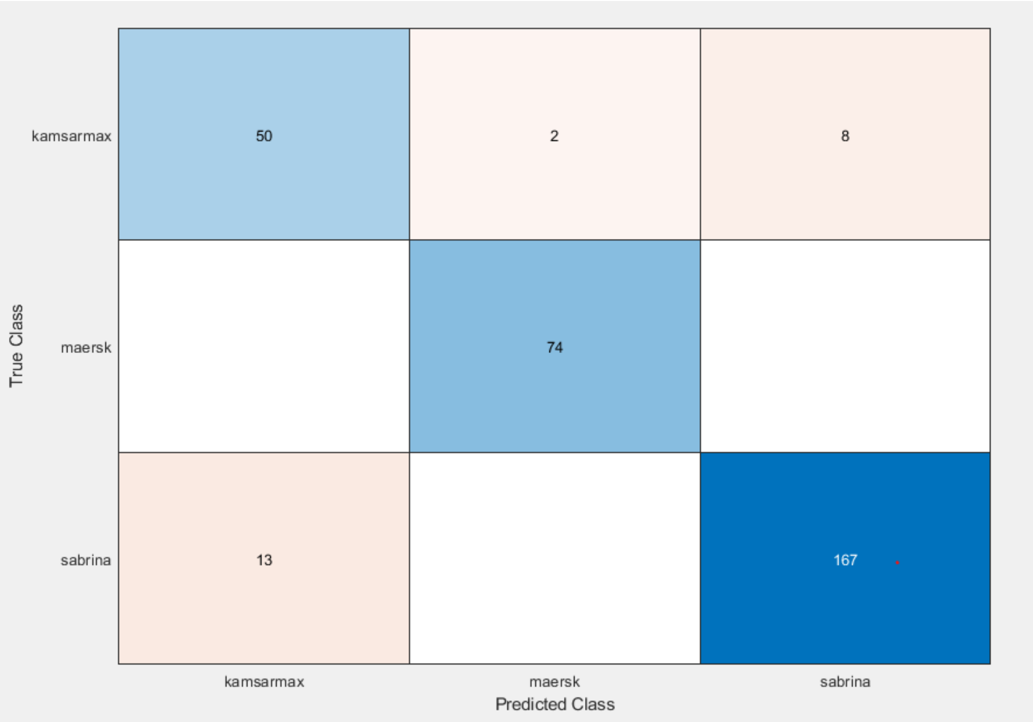


Figure 6.1: Confusion Matrix of Ship Detection System

6.3 Summary

This project focuses on the development of an advanced ship detection system designed for ISAR images, utilizing deep learning, computer vision, and fuzzy logic techniques. Despite limitations such as computational hurdles, constrained data availability, and performance challenges, the system demonstrates remarkable results and contributes to the field of ship detection. Potential improvements include expanding the dataset, enhancing the YOLOv7 neural architecture for the ship feature detector, and integrating individual FIS models into a more efficient model. Additionally, addressing real-time evaluation and hardware compatibility issues is essential. These advancements will enhance the system's robustness, accuracy, and real-world performance, driving progress in ship detection technology. The integration of ship classifier, ship feature detector, and FIS model into C/C++ software also presents challenges for future development.

REFERENCES

[1] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

[2] Redmon, J., Farhadi, A. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[3] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 580-587.

[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 91-99.

[5] Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 1097-1105.

[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.

[9] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[10] Chen, G., Bai, H., & Yuille, A. (2016). Training a convolutional neural network for multi-label classification with incomplete labeling. Proceedings of the IEEE conference on computer vision and pattern recognition, 1789-1798.

[11] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Retrieved from https://arxiv.org/abs/2004.10934.

[12] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2021). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence. doi: 10.1109/TPAMI.2021.3068772.

[13] Bezdek, J. C., Pal, S. K., & Abonyi, J. (2013). Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Springer.

[14] Chang, W. (2014). Fuzzy Systems Design Principles: Building Fuzzy If-Then Rule Bases. IEEE Transactions on Fuzzy Systems, 22(6), 1606-1619.

[15] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2014). Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer

[16] Mamdani, Ebrahim H., and Setrag N. Aslanian. "An experiment in linguistic synthesis with a fuzzy logic controller." International Journal of Man-Machine Studies 7.1 (1975): 1-13.

[17] W. Zhao, A. Heng, L. Rosenberg, S. T. Nguyen, L. Hamey and M. Orgun," ISAR Ship Classification Using Transfer Learning,", 2022 IEEE Radar Conference (RadarConf22), New York City, NY, USA, 2022, pp. 1-6, doi: 10.1109/RadarConf2248738.2022.9764304.

[18] L. Rosenberg, W. Zhao, A. Heng, N. Dahal, L. Hamey and M. Orgun," ISAR Ship Classification Using Metadata Features,", 2022 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, Australia, 2022, pp. 1-8, doi: 10.1109/DICTA56598.2022.10034611.

[19] Z. Yuan, Y. Jiang, J. Li, and H. Huang," Hybrid-DNNs: Hybrid Deep Neural Networks for Mixed Inputs", arXiv [cs.LG]. March, 2020, pp. 3-8, doi: 10.48550/arXiv.2005.08419.

[20] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020." Generalizing from a Few Examples: A Survey on Few-shot Learning". ACM Comput. Surv. 53, pp. 3, Article 63 May 2021.

[21] K. H. Brodersen, C. S. Ong, K. E. Stephan and J. M. Buhmann," The Balanced Accuracy and Its Posterior Distribution,", 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010, pp. 3121-3124, doi: 10.1109/ICPR.2010.764.

[22] Z. Wang, L. Du, J. Mao, B. Liu and D. Yang," SAR Target Detection Based on SSD With Data Augmentation and Transfer Learning,", in IEEE Geoscience and Remote Sensing Letters, vol. 16, no. 1, pp. 150-154, Jan. 2019, doi: 10.1109/LGRS.2018.2867242.

[23] S. Sharma, K. Senzaki and H. Aoki," Comparative Study of Feature Extraction Approaches for Ship Classification in Moderate-Resolution SAR Imagery," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 2018, pp. 6995-6998, doi: 10.1109/IGARSS.2018.8518966.