Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy , Thrissur,
Kerala 679531

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

Ship Detection Using SAR Imagery

PROJECT REPORT

NITHIN PETER (JEC17CS075)

JEVIN PAULY (JEC17CS055)

TESSA SHYJU (JEC17CS102)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Under the guidance of

Mr. ANIL ANTONY

CREATING TECHNOLOGY LEADERS OF TOMORROW

JUNE 2021

Department of Computer Science & Engineering

Jyothi Hills, Panjal Road,
Vettikattiri PO, Cheruthuruthy , Thrissur,
Kerala 679531

Jyothi Engineering College
NAAC Accredited College with NBA Accredited Programmes*
Approved by AICTE & affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777
NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

Ship Detection Using SAR Imagery

PROJECT REPORT

NITHIN PETER (JEC17CS075)
JEVIN PAULY (JEC17CS055)
TESSA SHYJU (JEC17CS102)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

of

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Under the guidance of

Mr. ANIL ANTONY

JUNE 2021

Department of Computer Science & Engineering

Department of Computer Science and Engineering

JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY

THRISSUR 679 531

JUNE 2021

BONAFIDE CERTIFICATE

This is to certify that the main project report entitled **Ship Detection Using SAR Imagery** submitted by **Nithin Peter (JEC17CS075), Tessa Shyju (JEC17CS102) and Jevin Pauly (JEC17CS055)** in partial fulfillment of the requirements for the award of **Bachelor of Technology** degree in **Computer Science and Engineering** of **A P J Abdul Kalam Technological University** is the bonafide work carried out by them under our supervision and guidance.

Mr. Anil Antony	**Dr. Swapna B Sasi**	**Dr. Saju P John**
Project Guide	Project Coordinator	Head of The Dept
Assistant Professor	Associate Professor	Associate Professor
Dept. of CSE	Dept. of CSE	Dept. of CSE

Jyothi Engineering College
NAAC Accredited college with NBA Accredited programmes*
Approved by AICTE & Affiliated to APJ Abdul Kalam Technological University
A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR
JYOTHI HILLS, VETTIKKATTIRI P.O., CHERUTHURUTHY, THRISSUR PIN-679531 PH: +91-4884-259000, 274423 FAX : 04884-274777
www.jecc.ac.in | E-mail : info@jecc.ac.in

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

COLLEGE VISION

Creating eminent and ethical leaders through quality professional education with emphasis on holistic excellence.

COLLEGE MISSION

- To emerge as an institution par excellence of global standards by imparting quality engineering and other professional programmes with state-of- the-art facilities.
- To equip the students with appropriate skills for a meaningful career in the global scenario.
- To inculcate ethical values among students and ignite their passion for holistic excellence through social initiatives.
- To participate in the development of society through technology incubation, entrepreneurship and industry interaction.

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

DEPARTMENT VISION

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

DEPARTMENT MISSION

- To create technically competent and ethically conscious graduates in the field of Computer Science & Engineering by encouraging holistic learning and excellence.
- To prepare students for careers in Industry, Academia and the Government.
- To instill Entrepreneurial Orientation and research motivation among the students of the department.
- To emerge as a leader in education in the region by encouraging teaching, learning, industry and societal connect

PROGRAMME OUTCOMES (POs)

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-Long Learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. The graduates shall have sound knowledge of Mathematics, Science, Engineering and Management to be able to offer practical software and hardware solutions for the problems of industry and society at large.
2. The graduates shall be able to establish themselves as practising professionals, researchers or Entrepreneurs in computer science or allied areas and shall also be able to pursue higher education in reputed institutes.
3. The graduates shall be able to communicate effectively and work in multidisciplinary teams with team spirit demonstrating value driven and ethical leadership.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. An ability to apply knowledge of data structures and algorithms appropriate to computational problems.
2. An ability to apply knowledge of operating systems, programming languages, data management, or networking principles to computational assignments.
3. An ability to apply design, development, maintenance or evaluation of software engineering principles in the construction of computer and software systems of varying complexity and quality.
4. An ability to understand concepts involved in modeling and design of computer science applications in a way that demonstrates comprehension of the fundamentals and tradeoffs involved in design choices.

COURSE OUTCOMES (COs)

C410.1 The students will be able to analyse a current topic of professional interest and present it before an audience.

C410.2 Students will be able to identify an engineering problem, analyse it and propose a work plan to solve it.

C410.3 Students will have gained thorough knowledge in design, implementations and execution of Computer science related projects.

C410.4 Students will have attained the practical knowledge of what they learned in theory subjects.

C410.5 Students will become familiar with usage of modern tools.

C410.6 Students will have ability to plan and work in a team.

ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all respected personalities who had guided, inspired and helped us in the successful completion of this interim project. First and foremost, we express my thanks to **The Lord Almighty** for guiding us in this endeavour and making it a success.

We take immense pleasure in thanking the **Management** of Jyothi Engineering College and **Dr. Sunny Joseph Kalayathankal**, principal, Jyothi Engineering College for having permitted me to carry out this seminar. Our sincere thanks to **Dr. Saju P John**, Head of the Department of Computer Science and Engineering for permitting us to make use of the facilities available in the department to carry out the interim project successfully.

We express our sincere gratitude to **Mr. Shaiju Paul & Dr. Swapna B Sasi**, project co-ordinators for their invaluable supervision and timely suggestions. We are very happy to express our deepest gratitude to our mentor **Mr. Anil Antony**, Assistant Professor,Department of Computer Science and Engineering, Jyothi Engineering College for his able guidance and continuous encouragement.

Last but not least, we extend our gratefulness to all teaching and non-teaching staff who directly or indirectly involved in the successful completion of this interim project work and to all friends who have patiently extended all sorts of help for accomplishing this undertaking.

ABSTRACT

Ship detection plays an important role in marine transportation, fishery management, and maritime disaster rescue. Nowadays, the current researches almost are focusing on improving detection accuracy while detection speed is neglected. However, it is also extraordinarily important to increase the ship detection speed, because it can provide real-time ocean observation and timely ship rescue. Boat identification is also a very important and challenging task in maritime traffic monitoring, the difficulty of this task lies in the accurate positioning and identification of relatively small boats in complex scenes. Synthetic aperture radar (SAR) imagery has been used as a promising data source for monitoring maritime activities, and its application for oil and ship detection has been the focus of many previous research studies. Many object detection methods ranging from traditional to deep learning approaches have been proposed. However, majority of them are computationally intensive and have accuracy problems. The huge volume of the remote sensing data also brings a challenge for real time object detection. YOLO is a network for object detection. The detection task consists in determining the location on the image where certain objects are present, and classifying those objects. Previous methods for this, like R-CNN and its variations,used a pipeline to perform this task in multiple steps. This may be slow to run and also hard to optimize because each individual component must be trained separately. YOLO does it all with one neural network. You Only Look Once (YOLO) can be optimised to be used for ship detection. This algorithm makes it possible to do real time ship detection in an efficient manner. We are proposing a new and better method for ship detection using YOLO.

Keywords - Detection, Identification, Synthetic aperture radar, You Only Look Once, Convolutional neural networks.

CONTENTS

List of Figures

List of Abbreviations

CNN	: *Convolutional Neural Network*
R – CNN	: *Recurrent Convolutional Neural Network*
CRNN	: *Convolutional Recurrent Neural Network*
YOLOV3	: *You Only Look Once v*3
RGB	: *Red – Green – Blue*
HSV	: *Hue – Saturation – Value*
UAV	: *Unmanned Aerial Vehicle*
YOLO	: *You Only Look Once*
SSD	: *Single Shot Object Detection*
SAR	: *Synthetic Aperture Radar*
FC	: *Fully Connected*

CHAPTER 1

INTRODUCTION

1.1 Overview

With the development of economic globalization, maritime transportation is becoming more and more frequent. Due to complex shipping scenes, boat crossing, fog shielding and other problems, traffic accidents are constantly occurring. Ship detection has great demands in civil and military fields.For example, in the civil field, ship detection can supervise transportation, marine traffics and illegal smuggling. In the military field, one can monitor for cross-border smuggling or other illegal behaviors. However, traditional ship detection is based on naked eyes monitoring, which causes huge labor costs.

Computer-aided detection method greatly saves the labor cost and improves detection efficiency at the same time. Nowadays, with the development of artificial intelligence technology, more and more experts begin to study the methods of ship detection based on data-driven and artificial intelligence. Probably a simple explanation to this fact comes from that artificial intelligence methods can automatically extract ship's features, avoiding the manual feature engineering of traditional methods, which greatly improves the detection efficiency.

Synthetic aperture radar (SAR), an all-weather and all-time microwave sensor, is one of the most important tools in remote sensing filed. Up to now, many scholars have proposed many SAR ship detection methods, which have greatly promoted the development of SAR image interpretation. According to my survey, in SAR ship detection field, many scholars are focusing on improving the accuracy of ship detection while the detection speed is neglected. In fact, it is also extraordinarily important to increase the ship detection speed, because it can provide real-time ocean observation and timely ship rescue.

Here we will look at various modifications that can be made to the YOLOv3 algorithm to optimize it for ship detection specifically. We will look at methods that not only improve the accuracy but also increase the speed of detection. [5]

1.2 Objectives

- Build a ship detection system for SAR images.
- Make a system that has high accuracy.
- Increase the speed of the system without sacrificing accuracy.

1.3 Organization of the project

The report is organised as follow:

- **Chapter 1:Introduction** Gives an introduction to "Ship Detection Using SAR Imagery".
- **Chapter 2:Literature Survey** Summarizes the various existiin techniques that helps in achieving the desired result.
- **Chapter 3: Problem Statement** Discuses about the need for the proposed system
- **Chapter 4:Project Management** Contains the effective project management model to be used for the project.
- **Chapter 5:Proposed System** Describes the various steps involved to produce this project.
- **Chapter 5:System Requirements & Specification**Describes the various technologies needed for implementation.
- **Chapter 6:Conclusion** Concludes with the future scope of implementation.
- **References** Includes the references for the project.

CHAPTER 2
LITERATURE SURVEY

2.1 Ship Detection: An Improved YOLOv3 Method

The main contributions of this method is to determine the anchor settings for the ship dataset by kmeans++ algorithm. Here we Design a convolutional neural network named Darknet-ship to solve the problem of excessive YOLOv3 parameters.Embedind the Squeeze-and-Excitation module in YOLOv3 to increases the network's ability to extract global features. [3]

2.1.1 Dimension Clusters

YOLOv3 introduces anchors, a set of initial candidate boxes with fixed width and height. The settings of anchors boxes affect the detection accuracy and speed. Kmeans algorithm is selected to conduct dimension clusters in YOLOv3. How-ever, the kmeans algorithm is sensitive to the initial points.Therefore, an improved clusters algorithm named kmeans++ is introduced to solve this problem. The distance function of K-means++ algorithm is defined as:

$$d(a,b) = 1 - IOU(a,b)$$

where a is the size of rectangular box, b is centroid of the rectangular box. The IOU function represents the overlapping ratio of two rectangular boxes, as shown in Fig. 3.1.

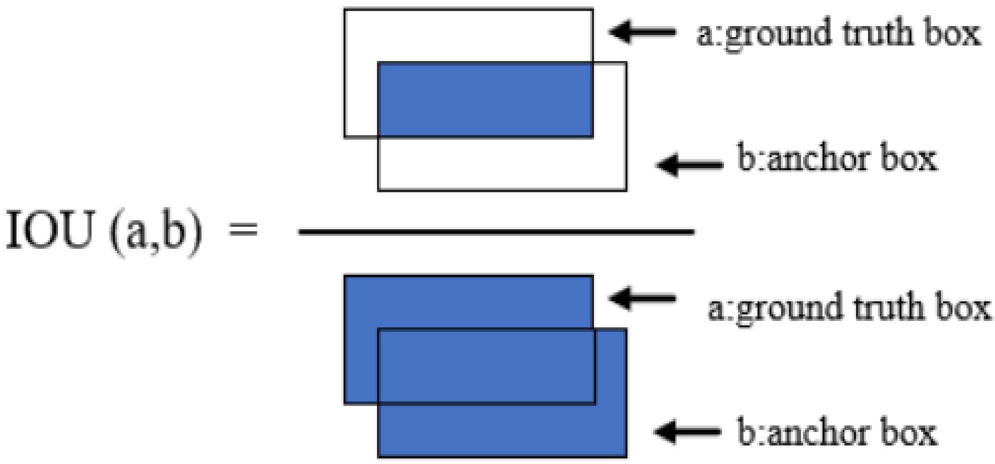


Figure 2.1: The description of IOU function

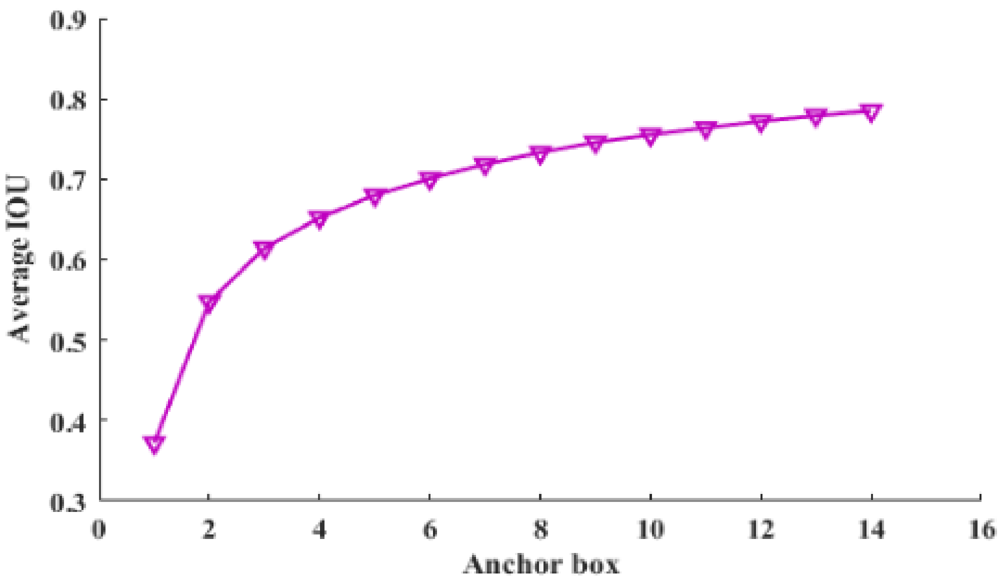


Figure 2.2: The relationship between the number of anchor boxes and average IOU

To determine the settings of anchor box, the relationship between average IOU and anchor box is depicted in Fig.3.2. According to the inflection point method, we selects six clusters and divide up the six clusters on three scales. the corresponding sizes of six clusters are: (31, 15), (65, 26), (115,42), (156, 28), (221, 55), (304, 104)

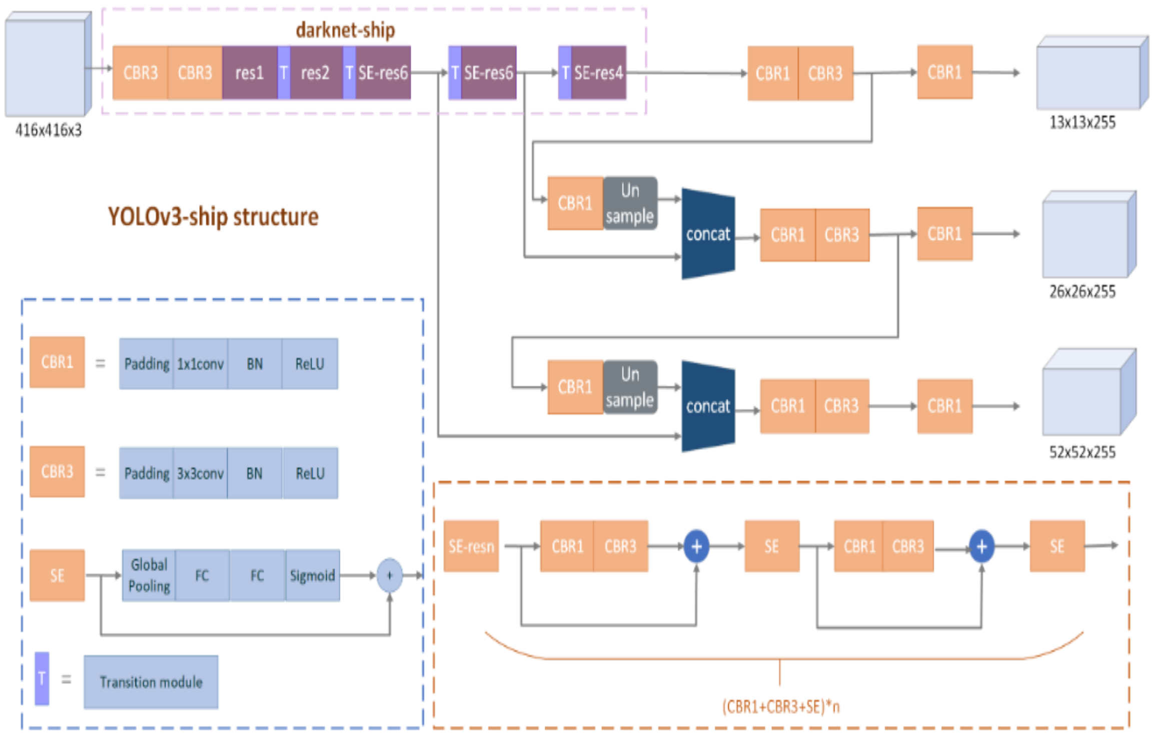


Figure 2.3: The description of YOLOv3-ship structure

2.1.2 Embedding of the Squeeze-and-Excitation Module

Squeeze-and-Excitation(SE) module is a ConvNet structure proposed by J. Hu in 2017, which won the championship of the imagenet classification competition. The SE module improves the expressive ability of network by accurately modeling the interaction between channels of convolution features.In the last three resnet blocks in Darknet-ship, the SE module is introduced to increase the receptive field and enhance the ability of network to extract global information. The SE module and the YOLOv3-ship structure are shown in Fig.3.5

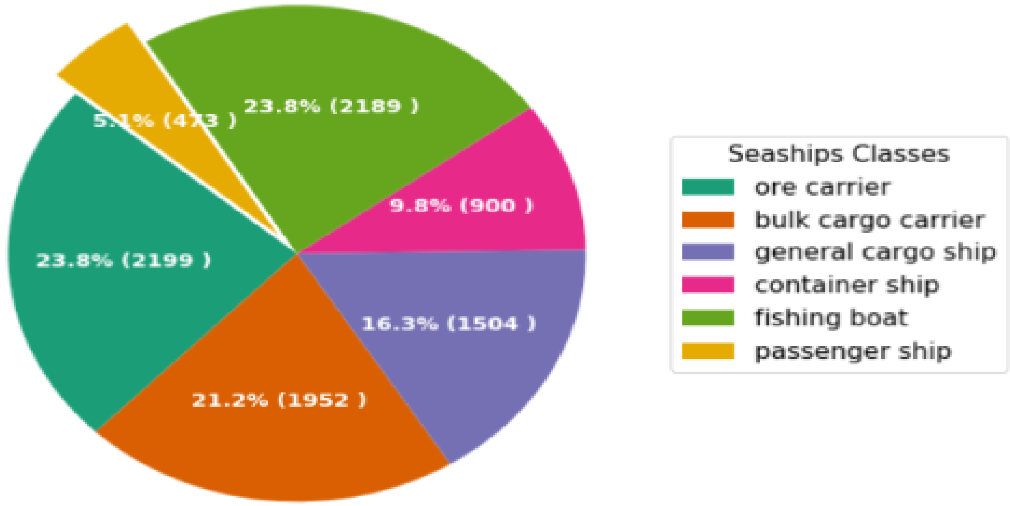


Figure 2.4: The numbers of bounding boxes for each class

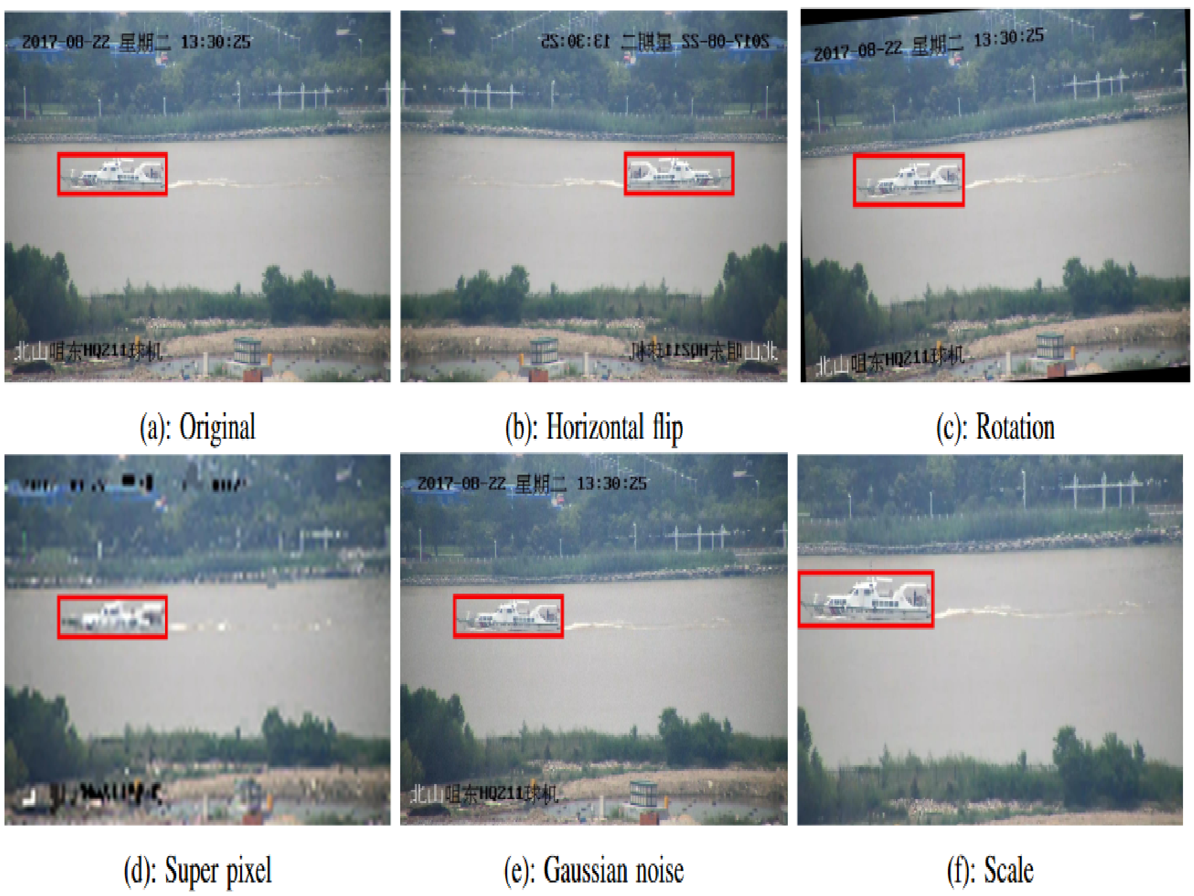


Figure 2.5: Some samples of the augmentation methods

2.2 High-Speed Ship Detection In SAR Images By Improved YOLOv3

Ship detection in synthetic aperture radar (SAR) images plays an important role in marine transportation, fishery management, and maritime disaster rescue. Synthetic aperture radar (SAR), an all-weather and all-time microwave sensor, is one of the most important tools in remote sensing filed. Up to now, many scholars have proposed many SAR ship detection methods, which have greatly promoted the development of SAR image interpretation. In this method we modify YOLOv3 to increase speed of detection without compromising the accuracy in ship detection from SAR images.

2.2.1 Improved Method

In order to improve the speed of ship detection, we have improved the original YOLOv3. Different from the original YOLOv3 where 20 types of targets need to be detected, SAR target detection this method contains only one class that is ship, so the reduction of network size does not significantly reduce accuracy by our research findings.The network structure of improved YOLOv3 is shown in fig.4.2. The detailed improvements are shown in the table1. From fig.4.2 and fig.4.3, to reduce the size of the network, we will use Darknet-19 as the backbone of the improved YOLOv3, which can reduce detection time. We also deleted repeated layers in YOLOv3-Scale1, YOLOv3-Scale2, and YOLOv3-Scale3. Finally, in order to make full use of the features extracted from the network, we have added two feature concatenation paths, which can improve the detection accuracy.[8]

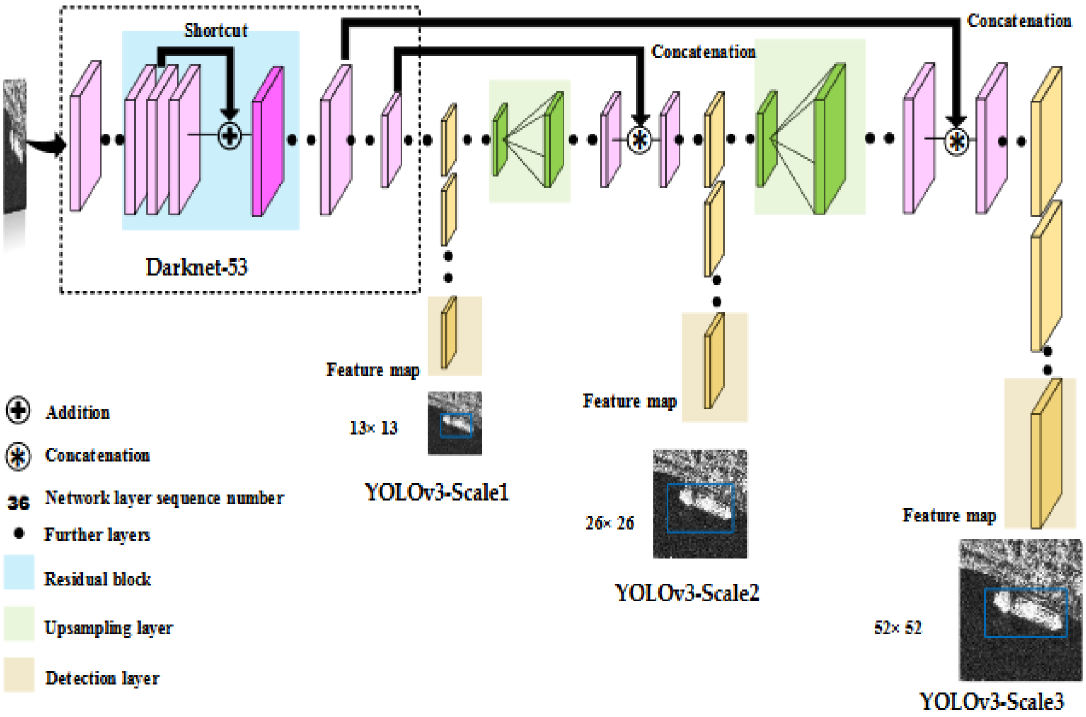


Figure 2.6: The network structure of original YOLOv3

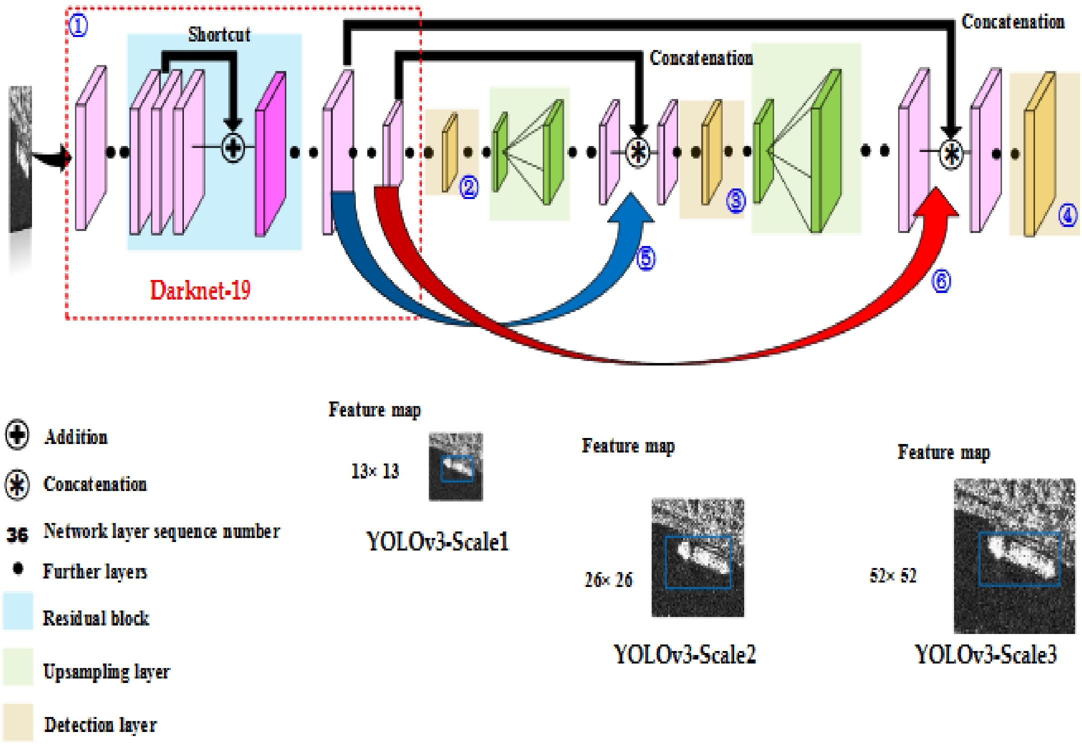


Figure 2.7: The network structure of improved YOLOv3

Number	Explanation
1	Change the backbone from Darknet-53 to Darknet-19.
2	Delete repeated layers of YOLOv3-Scale1.
3	Delete repeated layers of YOLOv3-Scale2.
4	Delete repeated layers of YOLOv3-Scale3.
5	Increase a concatenation.
6	Increase a concatenation.

Figure 2.8: The detailed improvements.

2.2.2 SAR Ship Detection Results

After some reasonable trainings, when the loss reaches the minimum, we get the final detection model. Then, we perform the actual SAR ship detection on the test set. The detection results of some samples are shown in the fig.4.4. From fig.4.4, almost all real ships can be detected correctly, and the original YOLOv3 and the improved YOLOv3 have similar detection accuracy.

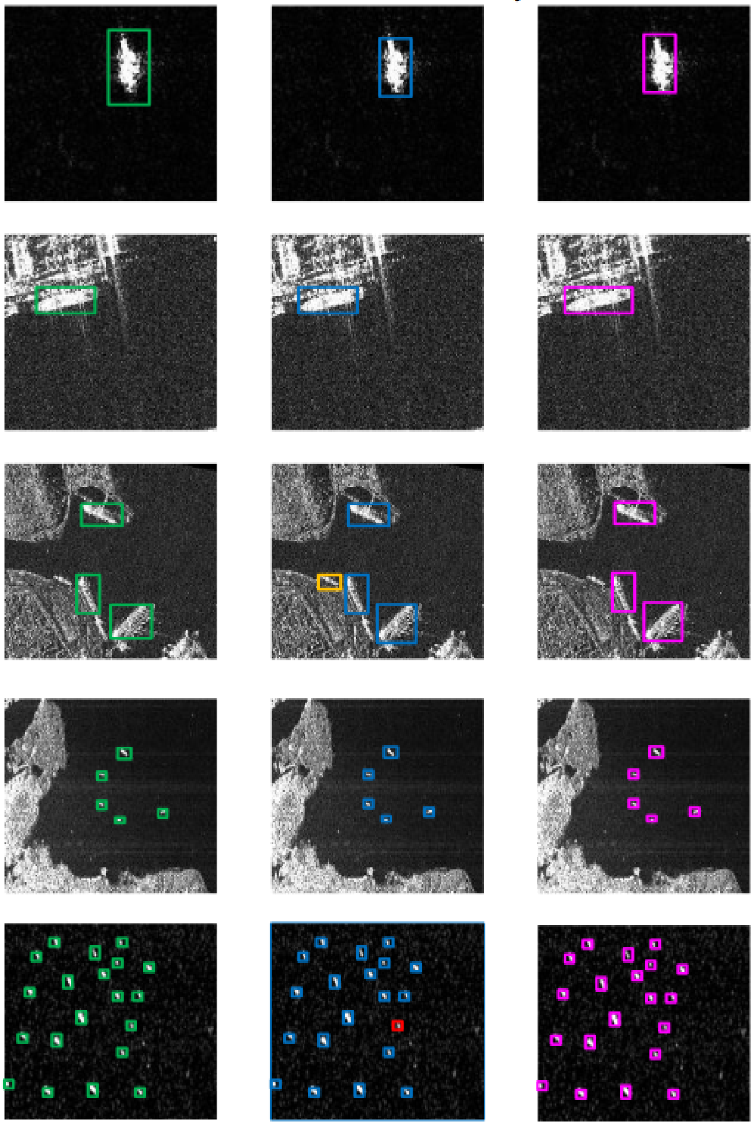


Figure 2.9: SAR ship detection results. (a) Ground truth. (b) Original YOLOv3. (c) Improved YOLOv3. Red is miss-detection and yellow is false alarm.

Fig.4.5 is the comparison of their quantitative evaluation indicators. From fig.3 and fig.4.5, the performance of the improved YOLOv3 is slightly better than the original YOLOv3, but the gap is narrow. More importantly, on the premise of keeping the accuracy basically unchanged, the improved network is smaller and lighter, which can reduce the detection time. In addition, due to the added two concatenation, the detection accuracy can remain unchanged. [8]

Method	Recall	Precision	mAP	F1-score
Original YOLOv3	89.12%	88.57%	90.14%	0.89
Improved YOLOv3	91.10%	89.36%	90.08%	0.90

Figure 2.10: Improvement Comparison.

2.3 Using convolutional neural network approach for ship detection in Senital-1 SAR imagery

In this method we use Convolutional Neural Networks (CNNs) method, the Faster R-CNN VGG16 in a SAR based image to detect the location of ships. This methodology is designed to detect and categorize the ships in congested areas of seaports where ships are close to each other.[6]

Conventional ship detection consists of the following main stages: land masking,pre-processing and discrimination. This study explores how faster R-CNN VGG16 neural network technique can be applied for ship detection and counting in congested areas, using SAR imagery. In first stage, pre-processing of SAR image is carried out. In pre-processing back-geocoding, averaging, binary conversion and image morphology are performed. After image morphology reflectance values of the images are normalized to a standard reflectance value. The convolutional neural network layers are used to classify the ships in SAR images.

2.3.1 PREPROCESSING OF SAR IMAGES

Sentinel-1A SAR images used in this study are single look complex images. Sub swaths of forty images are processed and back-geocoded. Back-geocoding is a digital elevational model (DEM) based coarse registration of images.

Figure 2.11: Single Back-coded SAR Image

A median of all these back-geocoded images is generated using this following formula.

(n + 1) ÷ 2th

where "n" is the number of items in the set and "th" just means the (n)th number.

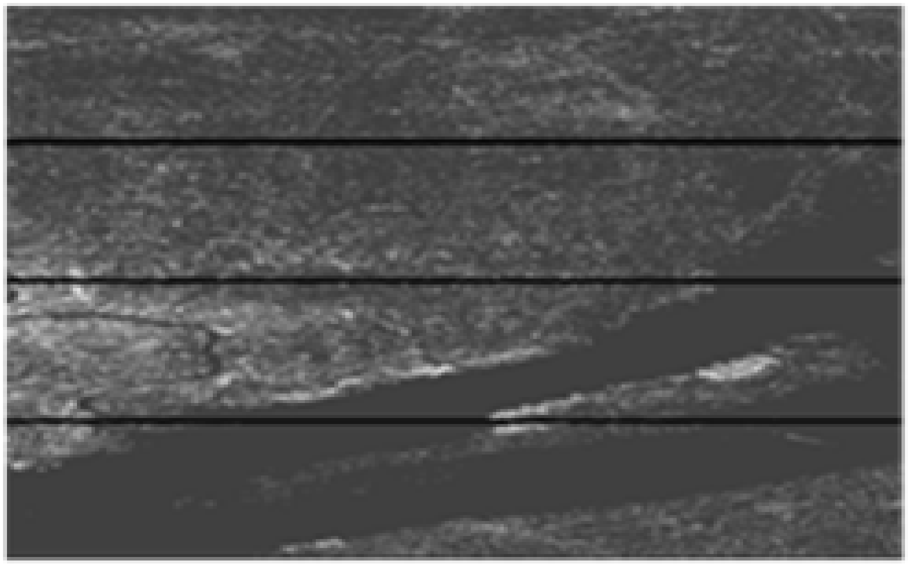


Figure 2.12: Median Image

Single back-geocoded images are differenced with median image in order to get a final image using following formula.

Differenced image = single image – median image

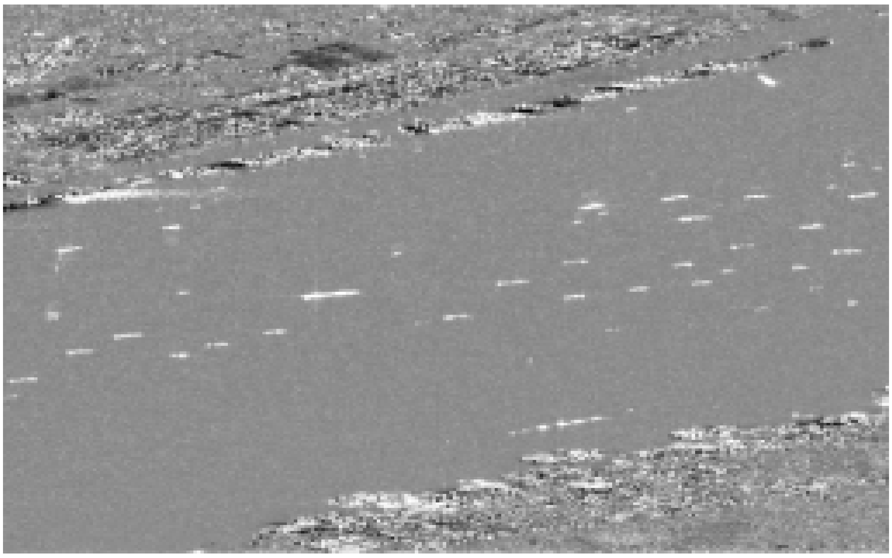


Figure 2.13: Difference Image

Each differenced image is then converted into binary. The image is then refined and performed adaptive adjustment to achieve the desired bright points on the black background by setting a threshold.

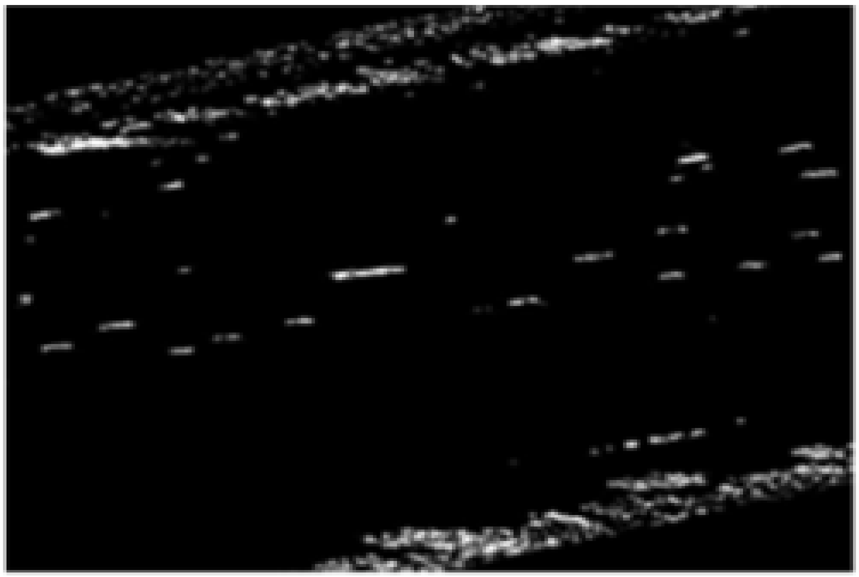


Figure 2.14: Binary Image With Specific Threshold

The elements on the surface are not very clear, because there are many holes in

it. In order to make the elements clearer and visible, a process of “dilation” is carried out. This process is a mathematical morphology operation that is used to enhance the pixel size in a binary image. Figure illustrates the process of dilation.

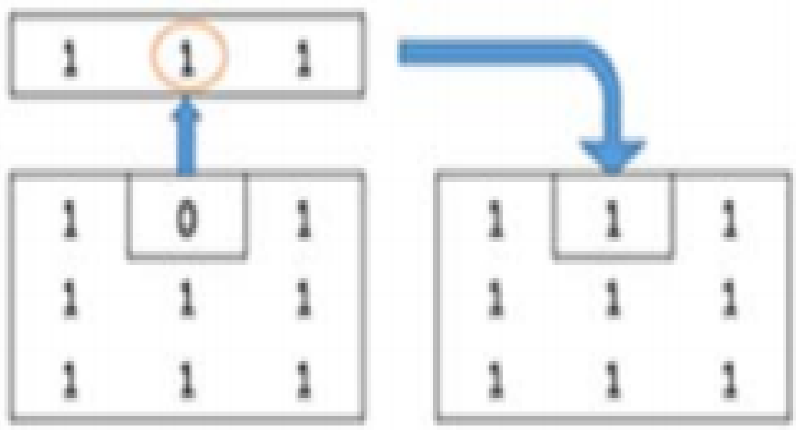


Figure 2.15: Dilation In Binary Image

This process uses the highest value of all the pixels in the neighbourhood as the value of the corresponding pixel in the output image. After that, the dilated image is further processed and a threshold is set to remove small elements in the image.

Figure 2.16: Dilated Image With Removed Small Objects

All the small holes are filled after the successful implementation of dilation process. This process will help in better identification and extraction of ship feature that will lead to the accurate detection of ship. After the process of dilation, the normalized value of pixel intensities in images are calculated to set a standard threshold.

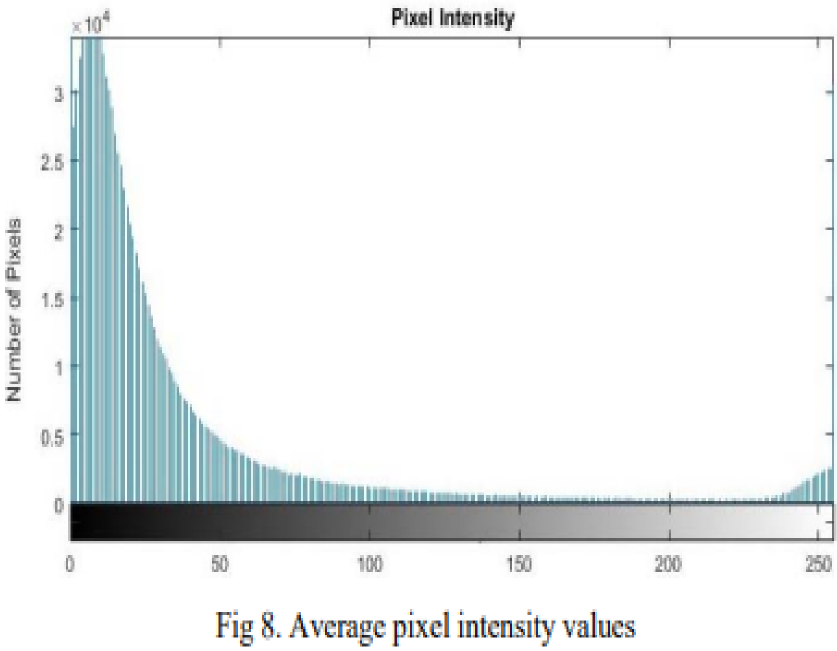


Figure 2.17: Average Pixel Intensity Values

Figure shows the excessive in concentration of black pixels however, we see number of bright intensities on the right side which depicts the brighter pixels of ships in the image. This pixel intensity is used to set a standard threshold for discriminating ships from other bright (grey or light-grey) pixels with less intensity.

2.3.2 TARGET CLASSIFICATION MODEL BASED ON CNN

:

In order to detect ships, we have to use R-CNN models. In this process of classification, an image passes through series of convolutional network layers, nonlinear, pooling and fully connected layer to get an output. The input images are passed through a previously trained CNN model and end up in convolutional feature map.Then it uses the features computed by CNN to locate and obtain predefined number of regions (bounding boxes) that may have objects. Finally R-CNN modules classify the elements in bounding boxes and adjust their coordinates. Ships are categorized into small medium and big.

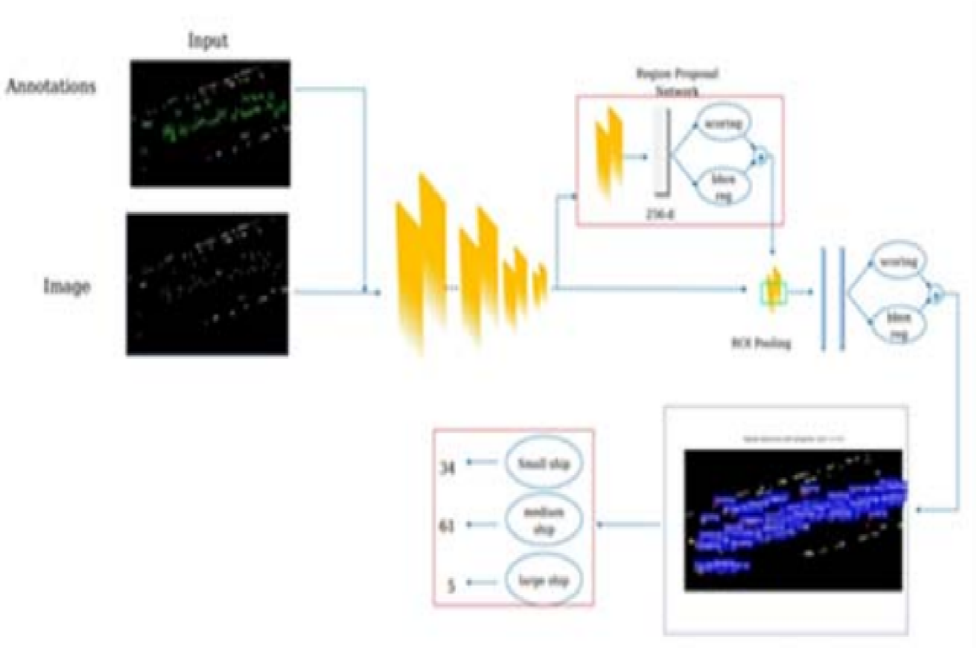


Figure 2.18: Ship Detection CNN Architecture

2.3.3 SHIP DETECTION

In the process of ship detection the image is classified into ship and sea. The image is further processed in convolutionary networks for extraction, selection and detection. In pre-processing stage back geocoding, averaging of image, image difference, thresholding and dilation are performed. Thus ship features are extracted. This is then passed through neural network layers and finally the ship is detected.

Figure 2.19: Ship Detection Process

2.4 Significance Based Ship Detection From SAR Imagery

Synthetic Aperture Radar images have potential applications in the surveillance scenario and hence automated target detection algorithms prove to be a useful tool in monitoring and crime control as well as in marine traffic management. The advancements in marine trade have lead to the increase in the number of ships in the world waters.Ships can be easily discerned in the SAR images due to their bright intensity which results due to the strong radar back scatter from their metal surface. These are the significant pixels in an image which can be gathered to detect the ship targets. During heavy sea state conditions and presence of speckle noise, sea ice and coastline structure, the ship detection process is affected since these non-ship features in the sea also exhibit high intensities in the SAR image.[1]

SAR images when acquired, suffer from patterns of constructive and destructive interference of backscattered signals from multiple distributed objects. This disturbance called as speckle noise being the largest source of noise in SAR images, results in bright and dark spots in the image, provoking complications in image interpretation, like failure to extract bright spots which can be possible targets or giving false alarms by identifying other disturbances as targets. Speckle noise is a locally correlated noise in multiplicative form due to which the image processing techniques sense great difficulties when applied on SAR imagery. The heterogeneity of sea clutter, different weather conditions and the unclear appearance of targets caused by different imaging angle in SAR imagery also adds to the difficulty in image interpretation [2]. In SAR oceanography, speckle noise is caused by the ripples and other objects on ocean that produces scattering. So the effect of speckle noise has to be reduced, in order to enhance the subsequent processes such as identification and characterization of the marine targets. A variety of methods have been attempted for despeckling by many researchers.

2.4.1 PROPOSED SHIP DETECTION ALGORITHM

A significance based ship detection method is proposed in this work as shown in Figure 2.5. The potential target pixels obtained by extracting the significant points in the image which are then discriminated as ship pixels and non-ship pixels using an ensemble classifier to improve the precision of the algorithm.

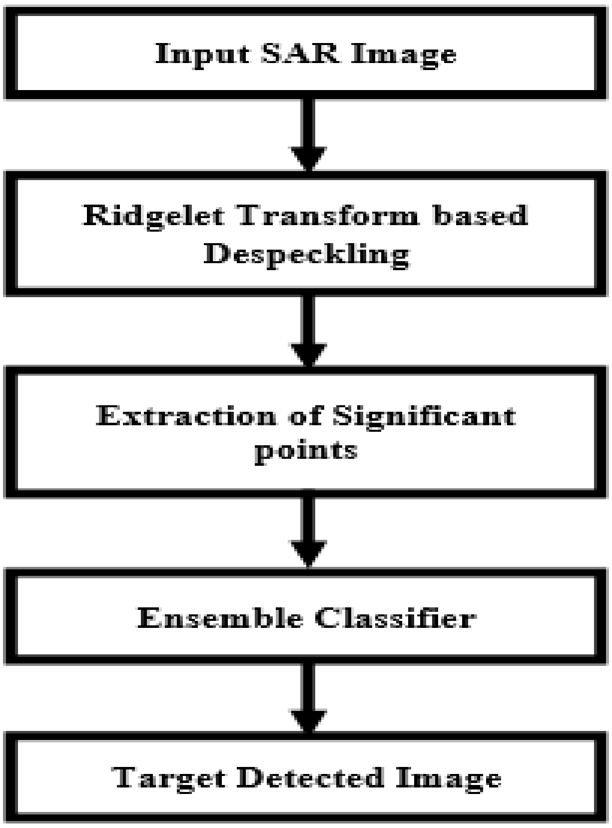


Figure 2.20: Proposed Target Detection Method

Adhering to the fact that, the ship pixels appear to be brighter than the background clutter pixels in the oceanographic SAR images, the target detection algorithm is proposed to capture the significant pixels as the potential target points. The background clutter, ocean waves, ship wakes and other structures in the ocean may also appear as bright points. So, an ensemble classifier is used to discriminate the ship and non-ship pixels based on their features. Prior to target detection, Ridgelet transform based despeckling is performed to enhance the further processing.

2.4.2 EXPERIMENTAL REULTS AND DISCUSSIONS

The experimentation of the proposed work was carried out using seven SENTINEL 1A SAR images. These are VV polarized, Ground Range Detected, Strip Map products acquired at high resolution mode. These images have 23 x 23 (rg x az) m resolution.

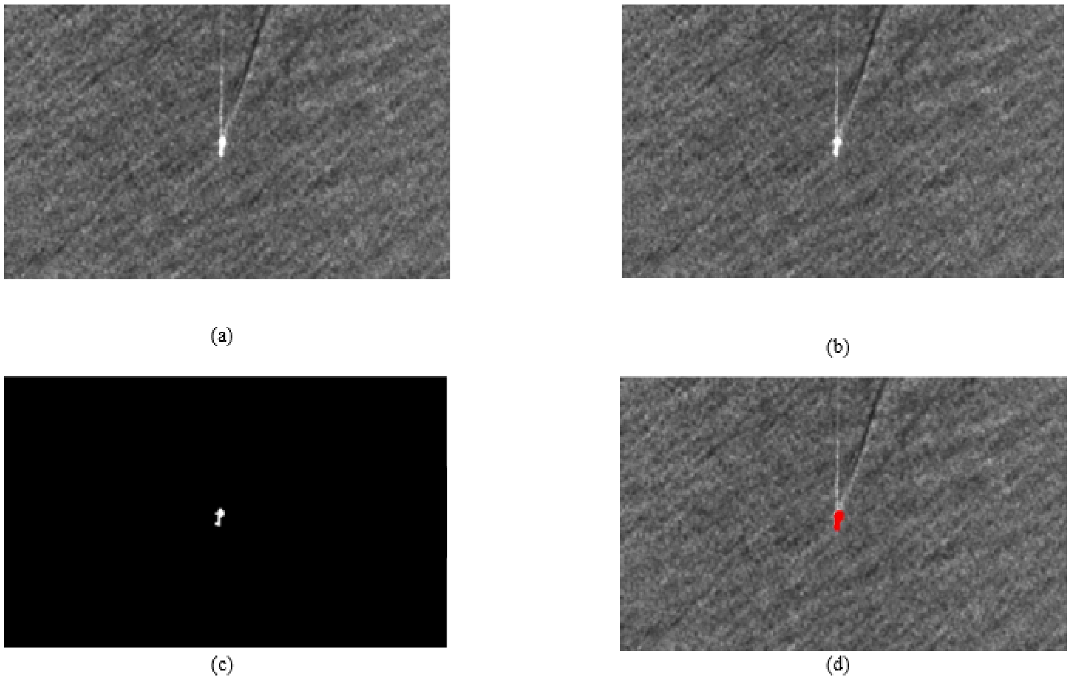


Figure 2.21: (a) Input SAR Patch; (b) Despeckled Patch; (c) Significance Patch; (d) Target Detected Patch

A significance based ship detection method is proposed in this work. The potential target pixels obtained by extracting the significant points in the image which are then discriminated as ship pixels and non-ship pixels using an ensemble classifier to improve the precision of the algorithm. Proposed Target Detection Method Adhering to the fact that, the ship pixels appear to be brighter than the background clutter pixels in the oceanographic SAR images, the target detection algorithm is proposed to capture the significant pixels as the potential target points. The background clutter, ocean waves, ship wakes and other structures in the ocean may also appear as bright points. So, an ensemble classifier is used to discriminate the ship and non-ship pixels based on their features. Prior to target detection, Ridgelet transform based despeckling is performed to enhance the further processing.

2.5 Classification of Patterns on High Resolution SAR Images

The SAR sensors are capable to operate in all types of weather conditions. Also they are capable to operate from very long ranges and also over wide areas of coverage. These features make them extremely attractive for monitoring the Earth's resources. We can analyze the presence of vegetation, mountains, oil sources, mineral resources etc. on a particular region even on other planets by pattern classification on SAR images. SAR images have achieved a prominent position in the field of remote sensing. Remote sensing is a means of acquiring information using airborne equipment and techniques to determine the characteristics of an area. Over the past few decades, SAR imaging has witnessed numerous advancements due to active research and development in it. These images will find many applications in resource management, agriculture, mineral exploration and environmental monitoring. But the effective use of SAR images requires an understanding of the nature and limitation of the data and of the various methods for processing the image and interpreting data from it.

The images obtained from radar contain noise, especially speckle noise. It adds to SAR images a granular aspect with random spatial variations. It degrades the quality of the image. The consequence with the speckle noise is that, the fine features of the SAR image could be lost while analyzing the image for certain purposes like pattern classification. Thus, the speckle noise present in SAR images hinders the proper interpretation of data and thereby makes the processes of segmentation, classification, and analysis of patterns in the image increasingly difficult and inaccurate. Thus we have to perform denoising prior to any processing in the SAR images. By denoising, the clarity of the data in the image gets enhanced and it helps in the proper interpretation of data.

Mainly all denoising techniques aim at better preserving image structures while reducing the noise. Once the image is denoised, we can perform pattern classification on it. Patterns can be anything like rivers, vegetation, urban areas etc. Pattern classification can be done with the help of some features that is obtained from the region of interest. These features are mainly obtained using certain feature extraction methods. Finally, the classification is done using proper classification algorithm that is having better classification accuracy and less time complexity. In this project, we are focusing on detecting different patterns like water regions, buildings and land areas and finally the farmlands and vegetation. [2]

2.5.1 Denoising

Denoising is done prior to feature extraction in the classification problem to remove the unwanted speckle noise present in the high resolution SAR images. The speckle noise adds to SAR images a granular aspect with random spatial variations. It degrades the quality of the image. The consequence with the speckle noise is that,the fine features of the SAR image could be lost while analyzing the image for certain purposes like pattern classification. This is because the presence of speckle noise reduces the possibilities of visual interpretation. Here, the denoising process is based on Maximum Likelihood estimation method using a robust mestimator known as Geman-McClure estimator. This method denoises the image without any loss in fine features or edge structures. Also, this method increases the accuracy of further processing on the SAR images.

Among the existing denoising technologies, NL means algorithm can be adaptively extended to get better results in denoising. This technology is based on the redundancy of neighboring patches in the image. In this technology, the noise free estimated value of a pixel is defined as a weighted mean of pixels in a certain region. The local and Non local ML estimation methods have some drawbacks. The local ML estimation causes blurring of edges and the distortion of fine structures in the image while the Non local ML estimation causes either under or over-smoothing. For denoising, maximum likelihood estimation along with the Geman-McClure estimator is used.

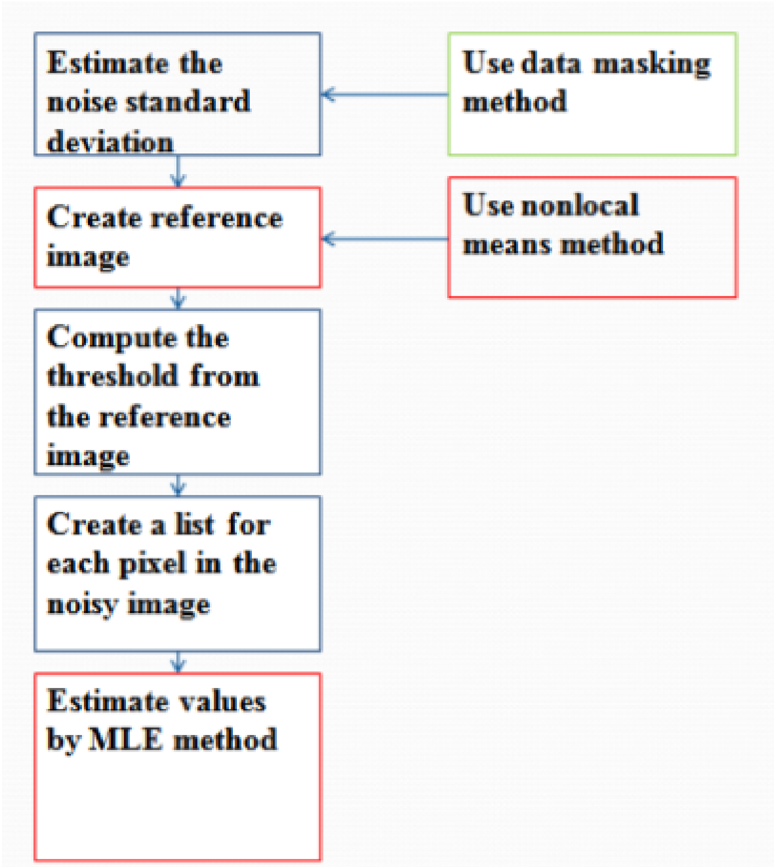


Figure 2.22: Block diagram for denoising

2.5.2 Estimation of noise standard deviation

From the noisy input image, estimation of noise standard deviation is performed by data masking technique. Spatial filtering is done in the conventional manner of convolving the image with a moving window called mask. In data masking, one method to estimate the noise standard deviation is to first filter the image to remove the image structure. By this, we get the image with only noise. For detecting edge structures Sobel edge detector is applied. Then the given image is subtracted from the Laplacian filtered image which results in the suppression of the image details.

2.5.3 Creating reference image

For creating reference image, Restricted Non Local Means method is used. Mainly the NL Means method was based on the Markovian hypothesis which states that pixels with a similar neighborhood have a similar gray level value. Here the filtered value at noisy pixel at a particular location is calculated using the NL Means method as a weighted average of all the pixels in the image.

$$v_i = \frac{1}{C_i} \Sigma_{j \varepsilon \Omega_s} w_{i,j} m_j \qquad (1)$$

where

$$C_i = \Sigma_{j \varepsilon \Omega_s} w_{i,j} \qquad (2)$$

This indicates normalization constant and the weight is given by the similarity of the Gaussian neighborhood between pixels and is expressed as

$$w_{i,j} = exp(-\frac{||N_i - N_j||^2_{2,a}}{h^2}) \qquad (3)$$

Because of the fact that exponential function degrades edge preserving in denoised image, our method uses Robust M-estimator function for weight calculation. Here we use Geman-McClure estimator function for calculating weight for Non Local Means method.

$$w(x,y) = \vartheta_{\sigma_s}(\Sigma_{t \in \Xi} GM_{\sigma_c}(t)(s(x+t) - (s(y+t))^2) \qquad (4)$$
$$:= \vartheta_{\sigma_s}(||s(x) - s(y)||_2, \sigma^2) \qquad (5)$$

After creating reference image, next we have to find the threshold value from it. This threshold value is used for further denoising process. On each 3X3 block of the reference image, we find the range value. Actually range value indicates the difference of largest value and smallest value in that particular block. Then, find the mode value of the range values obtained from the block. From this, we obtain the thresholds.

2.5.4 Maximum likelihood estimation

Before performing maximum likelihood estimation, we need to create list of pixels from the noisy image. Here we compare the pixel values in the noisy input image and pixel values in the noise free reference image. Then take the difference between the corresponding pixels in both the images using 3x3 block neighborhoods. Then the difference is compared with that of the threshold values obtained from reference image. Now those pixels in the noisy images with values less than the threshold will be used for denoising of the image using maximum likelihood estimation.

Algorithm 1 DENOISING ALGORITHM

Input: Noisy SAR image.
Output: Denoised image.

1. Estimate the noise standard deviation from the input
   magnitude using data masking technique
2 .Create the reference image using non local means method.
3. Compute the threshold t from the reference image by
   applying
   t=mode(range($v_f)_w$)
for every pixel m of M (input image) do **do**
   create the list,

$$l_i = m_j, (j\varepsilon\Omega_m)|abs(f(m_j) - f(m_i)) < t \quad (11)$$

   4. Using l_i values, estimate intensity values by maximum
      likelihood method
end for

2.5.5 Feature extraction

On the denoised high resolution SAR images, we could apply our fusion based method for the classification of target patterns. Here, we are considering pixel intensity difference by using Local Binary patterns (LBP). LBP is the particular case of the Texture Spectrum model. It has been found that Local Binary Pattern is a powerful feature for texture classification. By using LBP, we could distinguish homogeneous as well as heterogeneous areas in SAR images. The Local Binary pattern is based on the local relationships of pixel intensities. Image quanitization is done, matrix splitting is done to facilitate the movement of the window and connected components are traced out within the window.

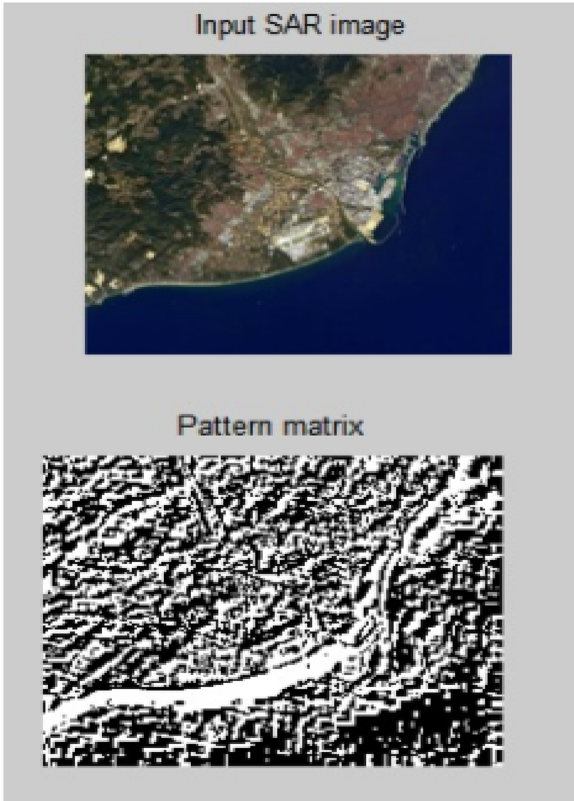


Figure 2.23: Input SAR image and its pattern matrix.

The second step is the splitting of the pattern matrix into three matrices: a 'positive matrix', a 'negative matrix' and an 'equal matrix' according to the values in the pattern matrix. Fig:5 shows the three matrices for a pattern matrix of an input SAR image. The three matrices play different roles in the image representation. The positive matrix captures brilliant patterns. That means points or regions which are significantly brighter than the central pixel. The negative matrix describes dark patterns, and the equal matrix captures a homogeneous regions.

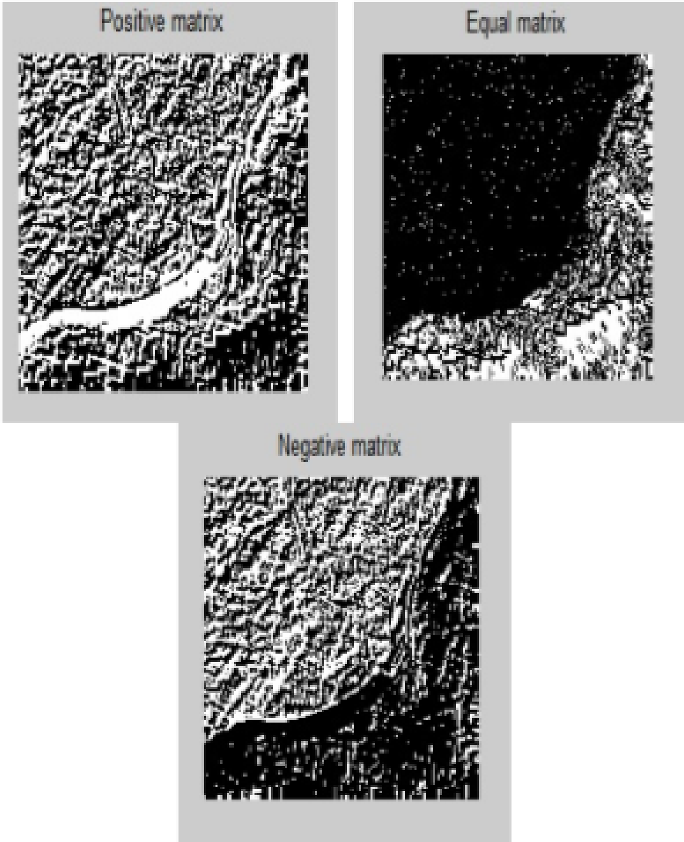


Figure 2.24: Pattern matrix of the input SAR image, positive matrix, equal matrix and negative matrix.

We are using color as a feature vector that is fed as input to the Artificial Neural Networks for classifying different patterns on SAR images. For that the neural network is trained using some known input-output samples based on color feature. That means we are training the ANN with the RGB values of different patterns on the SAR image. That means, the RGB combination of a particular pattern say,river has some uniqueness while comparing with other patterns. The values of R, G and B varies according to different patterns. We use this feature as input and provides output accordingly to the RGB value of the patterns.

2.5.6 Classification

Classification is the final stage by which different patterns like water region, farm lands and vegetation, buildings and urban areas etc get segmented. Here a fusion method is proposed for classifying these patterns on high resolution SAR images. We get segmented images of these patterns by using the three features LBP, HSV and RGB. Fig:4.4 shows the segmented images of water regions obtained by using LBP and HSV and the final segmented water region.

The results obtained from these methods are finally combined with the result obtained from RGB method together gives the final classification output. The output obtained from ANN classifier is again thresholded according to the target pattern.

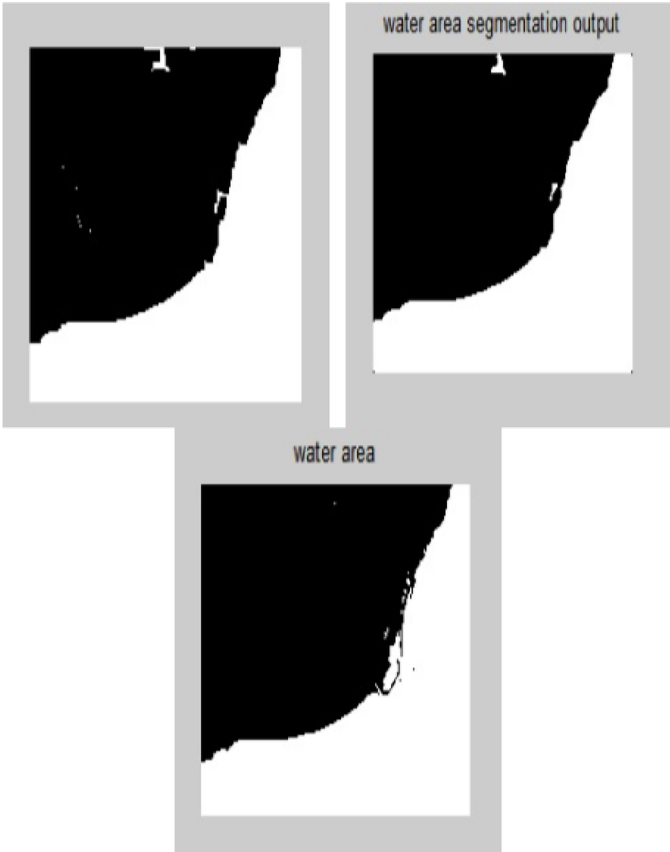


Figure 2.25: Segmented images obtained by using LBP, HSV and the final segmented water region.

Finally, we get the output image with patterns classified according to different colors where red indicates buildings and land areas, Green indicates vegetation and farm lands and blue indicates water regions. Fig:4.5 shows the final result of classification

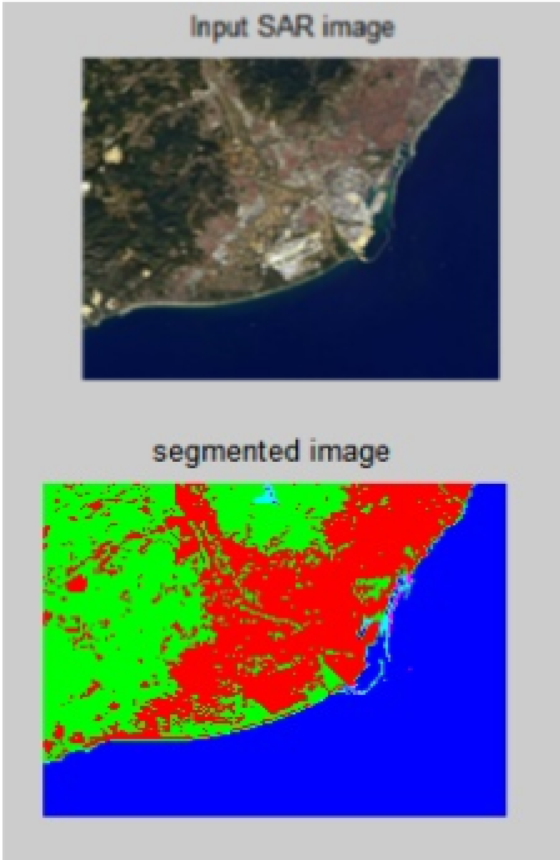


Figure 2.26: Input SAR image and its corresponding output image with patterns classified according to color.

CHAPTER 3
PROBLEM STATEMENT

Ship detection has great demands in civil and military fields.For example, in the civil field, ship detection can supervise transportation, marine traffic and illegal smuggling. In the military field, it can be used to monitor cross-border or other illegal behaviors. However, traditional ship detection is based on naked eyes monitoring, which causes huge labor costs. Computer-aided detection method greatly saves the labor cost and improves detection efficiency at the same time. Ship detection in synthetic aperture radar (SAR) images plays an important role in marine transportation, fishery management, and maritime disaster rescue. Nowadays, the current researches almost are focusing on improving detection accuracy while detection speed is neglected. However, it is also extraordinarily important to increase the ship detection speed, because it can provide real-time ocean observation and timely ship rescue. This project aims to provide a method for ship detection that is fast and also accurate. [4]

CHAPTER 4
PROJECT MANAGEMENT

4.1 Introduction

Project management is the discipline of planning, organizing, securing, managing, leading, and controlling resources to achieve specific goals. A project is a temporary endeavor with a defined beginning and end (usually time-constrained, and often constrained by funding or deliverables), undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with business as usual (or operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the management of these two systems is often quite different, and as such requires the development of distinct technical skills and management strategies.

In our project we are following the typical development phases of an engineering project

1. Initiation
2. Planning and Design
3. Execution and Construction
4. Monitoring and Controlling Systems
5. Completion

4.1.1 Initiation

The initiating processes determine the nature and scope of the project. The initiating stage should include a plan that encompasses the following areas :

1. Analysing the business needs/requirements in measurable goals
2. Reviewing of the current operations
3. Financial analysis of the costs and benefits including a budget
4. Stakeholder analysis, including users, and support personal for the project

5. Project charter including costs, tasks, deliverables, and schedule

4.1.2 Planing and design

After the initiation stage, the project is planned to an appropriate level of detail (see example of a flow-chart). The main purpose is to plan time, cost and resources adequately to estimate the work needed and to effectively manage risk during project execution. As with the initiation process, a failure to adequately plan greatly reduces the project's chances of successfully accomplishing its goals.

- Determining how to plan
- Developing the scope statement
- Selecting the planning team
- Identifying deliverables and creating the work breakdown structure
- Identifying the activities needed to complete those deliverables
- Developing the schedule
- Risk planning

4.1.3 Execution

Executing consists of the processes used to complete the work defined in the project plan to accomplish the project's requirements. The execution process involves coordinating people and resources, as well as integrating and performing the activities of the project in accordance with the project management plan. The deliverables are produced as outputs from the processes performed as defined in the project management plan and other frameworks that might be applicable to the type of project at hand.

4.1.4 Monitoring & controlling

Monitoring and controlling consists of those processes performed to observe project execution so that potential problems can be identified in a timely manner and corrective action can be taken, when necessary, to control the execution of the project. The key benefit is that project performance is observed and measured regularly to identify variances from the project management plan.

4.2 System Development Life Cycle

The Systems development life cycle (SDLC), or Software development process in systems engineering, information systems, and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system.

The SDLC phases serve as a programmatic guide to project activity and provide a flexible but consistent way to conduct projects to a depth matching the scope of the project. Each of the SDLC phase objectives is described in this section with key deliverables, a description of recommended tasks, and a summary of related control objectives for effective management. The project manager must establish and monitor control objectives during each SDLC phase while executing projects. Control objectives help to provide a clear statement of the desired result or purpose and should be used throughout the entire SDLC process.

4.2.1 Spiral Model

We have used the Spiral model in our project. The Spiral model incorporates the best characteristics of both- waterfall and prototyping model. In addition, the Spiral model also contains a new component called Risk Analysis, which is not there in the waterfall and prototype model. In the Spiral model, the basic structure of the software product is developed first. After the basic structure is developed, new features such as user interface and data administration are added to the existing software product. This functionality of the Spiral model is similar to a spiral where the circles of the spiral increase in diameter. Each circle represents a more complete version of the software product. The spiral is a risk-reduction oriented model that breaks a software project up into main projects, each addressing one or major risks. After major risks have been addressed the spiral model terminates as a waterfall model. Spiral iteration involves six steps:

1. Determine objectives, alternatives and constraints.
2. Identify and resolve risks.
3. Evaluate alternatives.
4. Develop the deliverables for the iteration and verify that they are correct.
5. Plan the next iteration.

6. Commit to an approach for the next iteration.

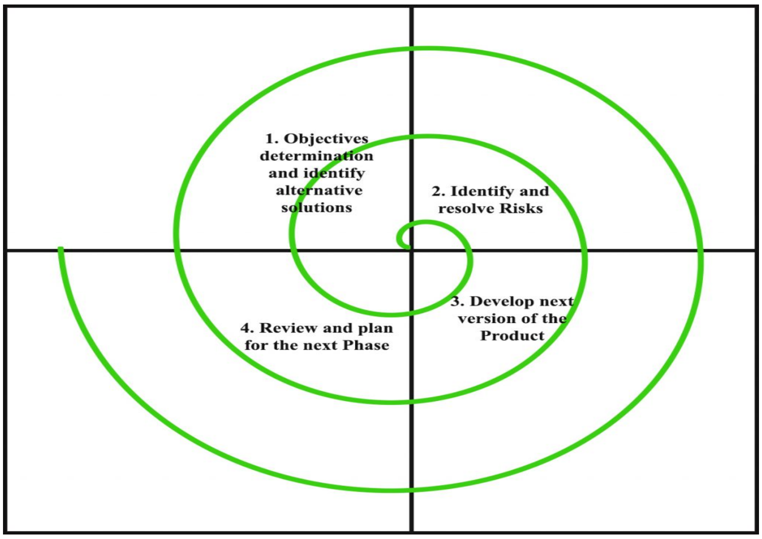


Figure 4.1: Spiral Model

CHAPTER 5
METHODOLOGY

5.1 System Requirements & Specifications

5.1.1 Windows 10

Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1 and was released to manufacturing on July 15, 2015, and to retail on July 29, 2015. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users. There are test builds of Windows 10 available to Windows Insiders. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

5.1.2 Python 3.6.2

Python is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

Python runs on Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines. Python is distributed under an OSI-approved open source license that makes it free to use, even for commercial products.

5.1.3 TensorFlow

Created by the Google Brain team, TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end

API for building applications with the framework, while executing those applications in high-performance C++. TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

5.1.4 Jupyter Environment

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

5.2 Proposed System

Modules

5.2.1 Data Acquisition Module

A large number of SAR images of ships are collected. In order to make the model as versatile as possible we use SAR images that come from different satellites, different polarization modes and different resolutions. Moreover, the backgrounds of ships are various. This ensures that the dataset is diverse and that the model can be used in various settings and can detect ships that are in various different backgrounds and locations. The annotations for the images are converted into YOLO format so that it becomes compatible with our model.

Here the HRSID Database is used. High resolution sar images dataset (HRSID) is a data set for ship detection, semantic segmentation, and instance segmentation tasks in high-resolution SAR images. This dataset contains a total of 5604 high-resolution SAR images and 16951 ship instances. HRSID dataset draws on the construction process of the Microsoft Common Objects in Context (COCO) datasets, including SAR images with different resolutions, polarizations, sea conditions, sea areas, and coastal ports. This dataset is a benchmark for researchers to evaluate their approaches. For HRSID, the resolution of SAR images is as follows: 0.5m, 1 m, and 3 m. [7]

In order to evaluate the effect of different methodological choices, we prepared three standard data sets for training and prediction:
D1: The standard training gives the model an insight into different types of scenarios where ships can be detected and identified. For This we used 70% of the images
D2:The prediction set or validation set includes the rest 20 % of the training set.
D3: The test set will be the overall leftover 10 % of the dataset, which we set aside for the final evaluation.

5.2.2 Image Enhancement Module

The SAR images like any other radar technology will have a lot of noise. Synthetic Aperture Radar (SAR) images are strongly corrupted by the speckle noise due to random electromagnetic waves interference. The speckle noise reduces the quality of images and makes their interpretation and analysis really difficult, so it's necessary to filter images to remove the noise in order to preserve as much as possible the most important features of the signal. It is important to remove this noise before these images are used for ship detection because they may cause variations in our model and training. So In this module we will remove that unwanted noise.

We use the HRSID database for our project. Initially the database was in the MS COCO format, that is the annotations were in the form of a Json file. We converted this file to the YOLO format. In the YOLO format each image has a text file containing the annotations. The text file will have the same name as the image file. All the images in the database were then resized to 416x416 to make the training easier. YOLO can be best trained when the size of the image is a multiple of 32.

Various image augmentation steps were added to vary the images slightly. We added +/-5 % noise , +/- 3% brightness and +/-3% contrast. Other augmentation types like reshape and mosaic were avoided as those kinds of images will not be produced in SAR and also they had a negative impact on the performance of the model.

5.2.3 Ship Detection Module

This module identifies the presence and location of the given SAR images. This is done with the help of YOLO algorithm. YOLO uses a totally different approach. YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance.

This model uses YOLOv5 that is the latest version of YOLO. YOLOv5 has multiple versions available, of which here the smallest model is used which is YOLOv5s. This verion is also the fastest and more than sufficient as there is only one class that we need to identify which is ship.

The HRSID database contains more than 5000 images. After we applied the augmentation steps the total number of images came up to more than 15000. We trained our model with 70% of the images. We then reduced the size of each layer and also the number of layers so as to prevent the model from overfitting. This also increased the efficiency of the model. The model depth multiple was set to 0.32 and the layer channel multiple was set to 0.24. It reduced the GPU usage, CPU usage and memory usage. This helped increase the accuracy of the model and make the model much faster. The model was trained only for 60 epochs. The best result was obtained in epoch 33 but we let it run a little longer to see if a better result could be obtained.

The size and location of the ship is calculated using the bounding box details produced by the model after it is used for detection. The size is calculated in terms of the number of pixels covered by the ship. The location of the ship is calculated by assuming a coordinate for the bottom left corner and top right corner of the image. This is used to calculate the exact location of all the ships identified in the image.

5.3 Data Flow Diagrams

5.3.1 Data Flow Diagram- Level 0

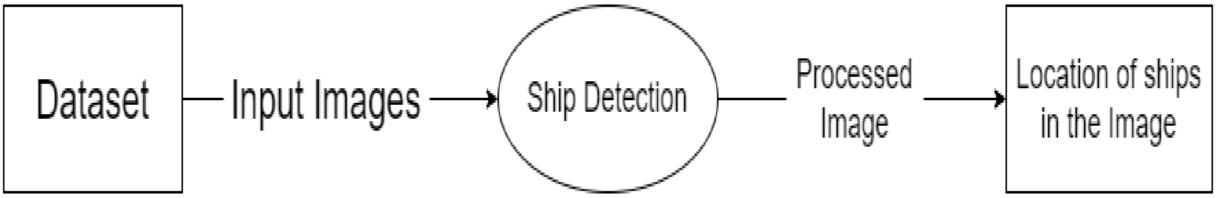


Figure 5.1: DFD- Level 0

5.3.2 Data Flow Diagram- Level 1

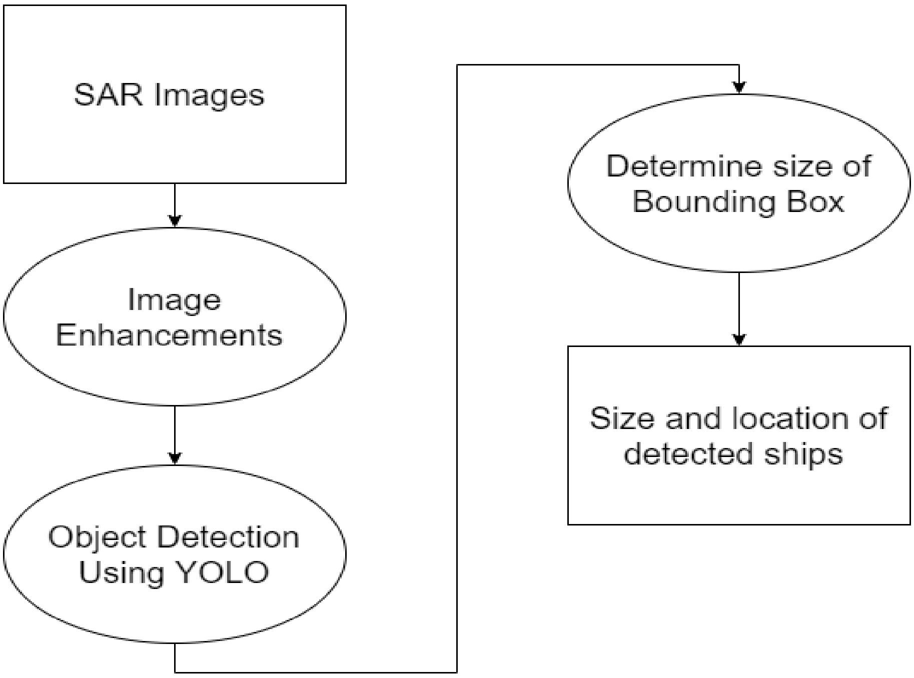


Figure 5.2: DFD- Level 1

5.3.3 Data Flow Diagram- Level 2

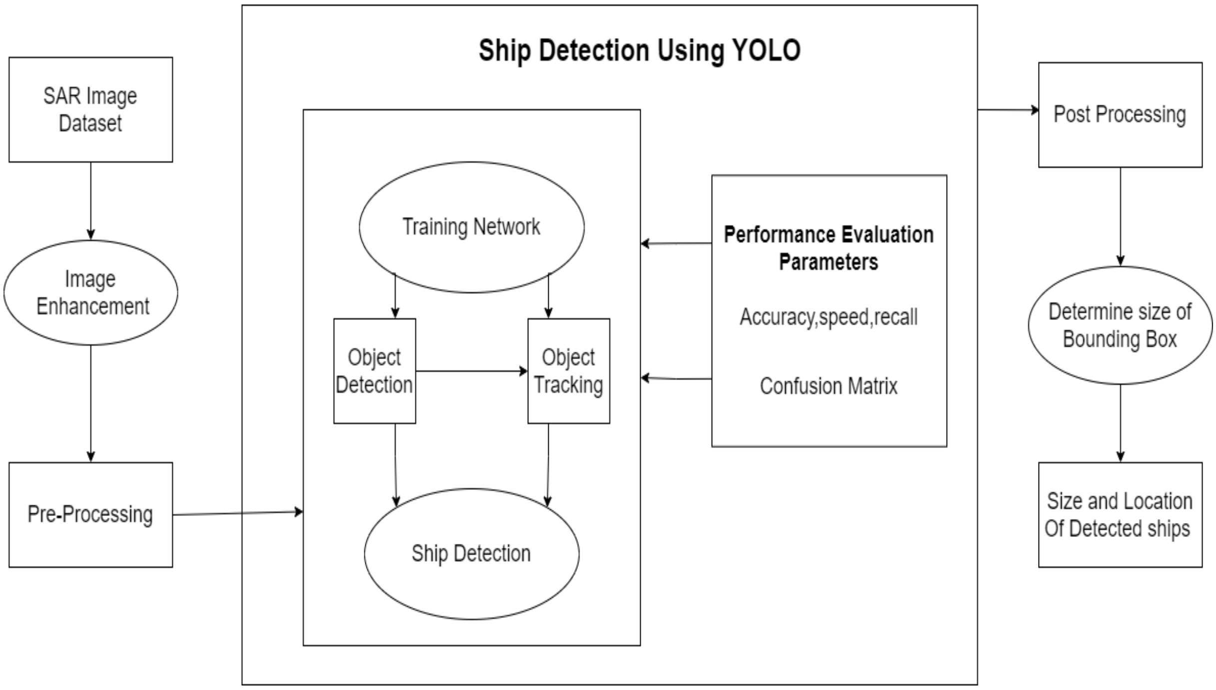


Figure 5.3: DFD- Level 2

5.4 Flow Chart

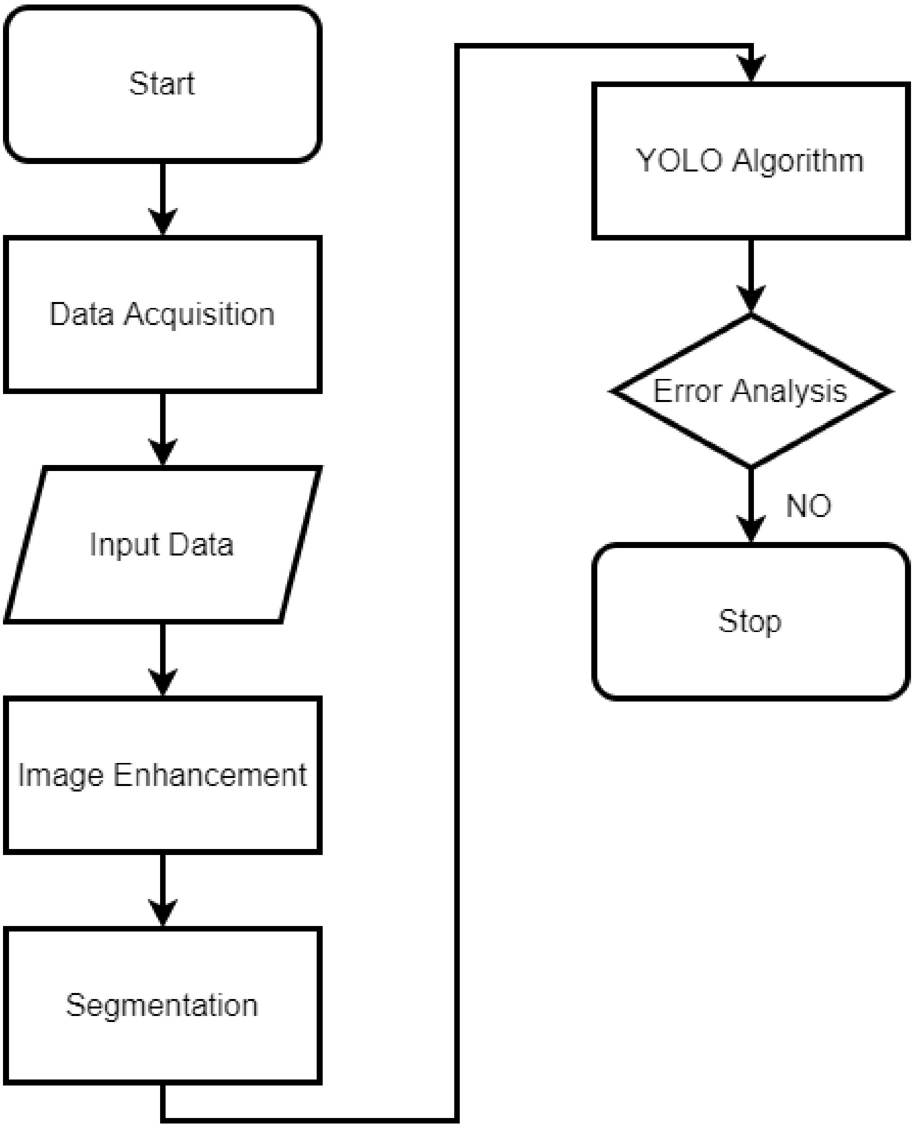


Figure 5.4: Flow Chart

5.5 Architecture

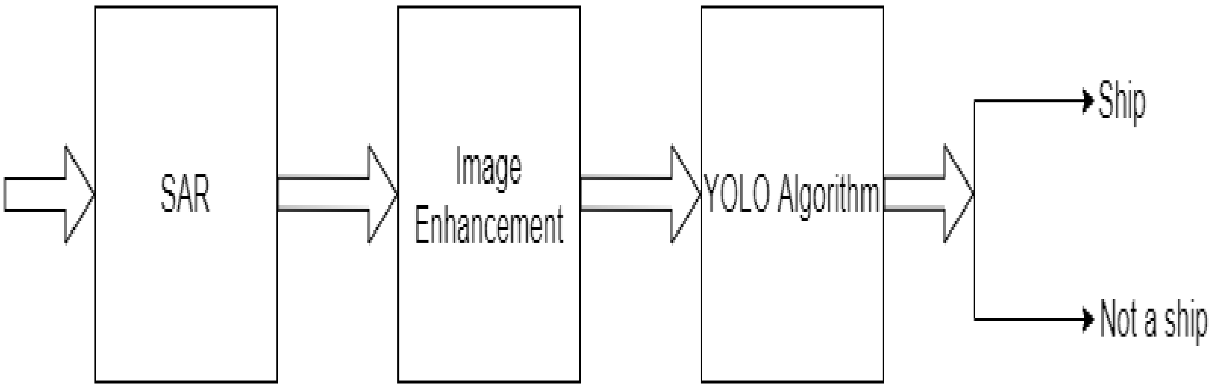


Figure 5.5: Architecture

CHAPTER 6

RESULTS

6.1 Training and Validation Set Results

With each consecutive Epoch the precision, recall and mean average precision slowly increases. To avoid over-fitting 60 Epochs were used. the model got a precision of 93.6%. The best thing is it only takes an average of 10ms to detect ships in an image. This shows that the model is both fast and accurate. The model is better than other similar models in both respects.

```
# train yolov5s on custom data for 100 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 60 --data '../data.yaml' --cfg ./models/custom_yolov5s.yaml --weights '' --name yolo

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     33/59    0.721G   0.01916  0.006185         0   0.02534         3       416: 100% 736/736 [01:00<00:00, 12.10it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 35/35 [00:04<00:00,  7.34it/s]
                 all  1.12e+03  3.31e+03     0.936      0.44     0.531       0.341

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     34/59    0.721G   0.01863  0.006112         0   0.02475         3       416: 100% 736/736 [01:00<00:00, 12.11it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 35/35 [00:04<00:00,  7.46it/s]
                 all  1.12e+03  3.31e+03     0.921     0.431     0.512       0.333

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     35/59    0.721G   0.01801  0.006029         0   0.02404         3       416: 100% 736/736 [01:00<00:00, 12.13it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 35/35 [00:04<00:00,  7.59it/s]
                 all  1.12e+03  3.31e+03     0.924     0.482     0.552       0.354

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     36/59    0.721G   0.01807  0.005951         0   0.02402         3       416: 100% 736/736 [01:00<00:00, 12.07it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100% 35/35 [00:04<00:00,  7.66it/s]
                 all  1.12e+03  3.31e+03     0.844     0.578     0.615       0.379
```

Figure 6.1: Training and Validation Set Result per Epoch

6.2 Testing Set Results

```
# use the best weights
%cd /content/yolov5/
!python detect.py --weights runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.4 --source ../test/images --save-txt
lov5/../test/images/P0014_4200_5000_2400_3200_jpg.rf.ce9e3dd5b405a6a7e0196d4075e15c16.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0014_4560_5360_6600_7400_jpg.rf.cf88da3124413ddf270f47fd795d4ef8.jpg: 416x416 1 ship, Done. (0.007s)
lov5/../test/images/P0015_0_800_7200_8000_jpg.rf.2ffb573e32947e2ff9c159fd29ecb15b.jpg: 416x416 7 ships, Done. (0.006s)
lov5/../test/images/P0015_0_800_7800_8600_jpg.rf.191da564f7cc6beb2aa64c626c6e1b43.jpg: 416x416 2 ships, Done. (0.006s)
lov5/../test/images/P0015_2400_3200_3000_3800_jpg.rf.b09fa5f734469af62d2f18908a428fc8.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0015_3000_3800_4200_5000_jpg.rf.2d4b264c3e59a61e4f830ffcbb017dce.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0015_4800_5600_2400_3200_jpg.rf.8823195a219d714a276969d240879764.jpg: 416x416 2 ships, Done. (0.006s)
lov5/../test/images/P0016_1200_2000_4200_5000_jpg.rf.5808e6aa83df5e18ff637973a83159d0.jpg: 416x416 2 ships, Done. (0.007s)
lov5/../test/images/P0016_3000_3800_0_800_jpg.rf.d9a5551a24f24ea68c3f151139f0057d.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0016_3000_3800_600_1400_jpg.rf.c367541b029ff6b3231ce437ee5faab5.jpg: 416x416 2 ships, Done. (0.007s)
lov5/../test/images/P0016_4910_5710_0_800_jpg.rf.cd14a81fa281e4c8b7759c2f9425971d.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0017_1200_2000_5400_6200_jpg.rf.e149592bfc83ed3464115ed699c62cda.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0017_2400_3200_1800_2600_jpg.rf.e212518b0e17a3d2aca82b6f8dda5754.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0017_3000_3800_1800_2600_jpg.rf.4cc5b02a508e811a2f93212a3d43d277.jpg: 416x416 2 ships, Done. (0.006s)
lov5/../test/images/P0017_4800_5600_3000_3800_jpg.rf.42e0a74b3971b3d4b6876966637c6f73.jpg: 416x416 1 ship, Done. (0.007s)
lov5/../test/images/P0017_4800_5600_7200_8000_jpg.rf.eddebe4d8697f8c7476d562eb0e18d83.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0017_600_1400_7800_8600_jpg.rf.64d7cd92ead73c0fac1ff3b833417265.jpg: 416x416 2 ships, Done. (0.006s)
lov5/../test/images/P0018_1200_2000_3000_3800_jpg.rf.53d85791438373a5584642d6fc1705aa.jpg: 416x416 1 ship, Done. (0.006s)
lov5/../test/images/P0018_1800_2600_4800_5600_jpg.rf.77d982a62deff05308f7c6b610e47aa7.jpg: 416x416 2 ships, Done. (0.006s)
olov5/../test/images/P0018_4800_5600_6600_7400_jpg.rf.66e7cb961bc52cdb867bc7357f43e294.jpg: 416x416 2 ships, Done. (0.006s)
olov5/../test/images/P0018_600_1400_8889_9689_jpg.rf.4f9c0cc03132e1216ef1ceff3a36f1e9.jpg: 416x416 1 ship, Done. (0.006s)
olov5/../test/images/P0019_1200_2000_7800_8600_jpg.rf.7ceb4d5399b2b73eb6eddc0e4be6deb5.jpg: 416x416 1 ship, Done. (0.006s)
olov5/../test/images/P0019_4200_5000_1800_2600_jpg.rf.4118376df2af29917e69eadda7728167.jpg: 416x416 1 ship, Done. (0.006s)
```

Figure 6.2: Testing Set Result

6.3 Graphical Representation of Result

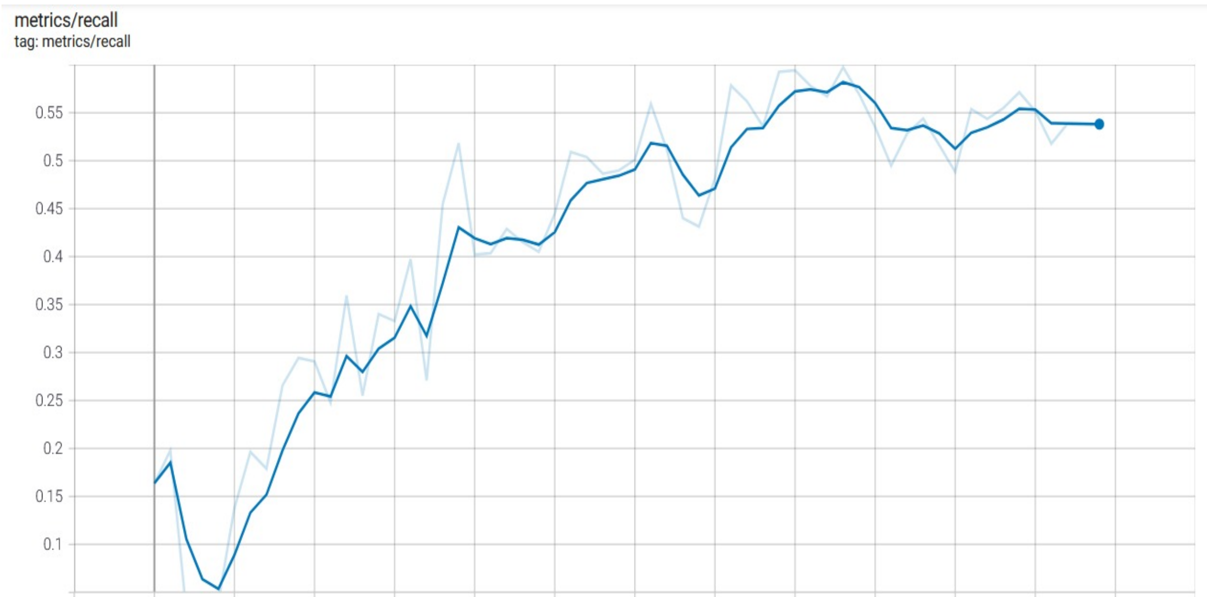


Figure 6.3: Recall

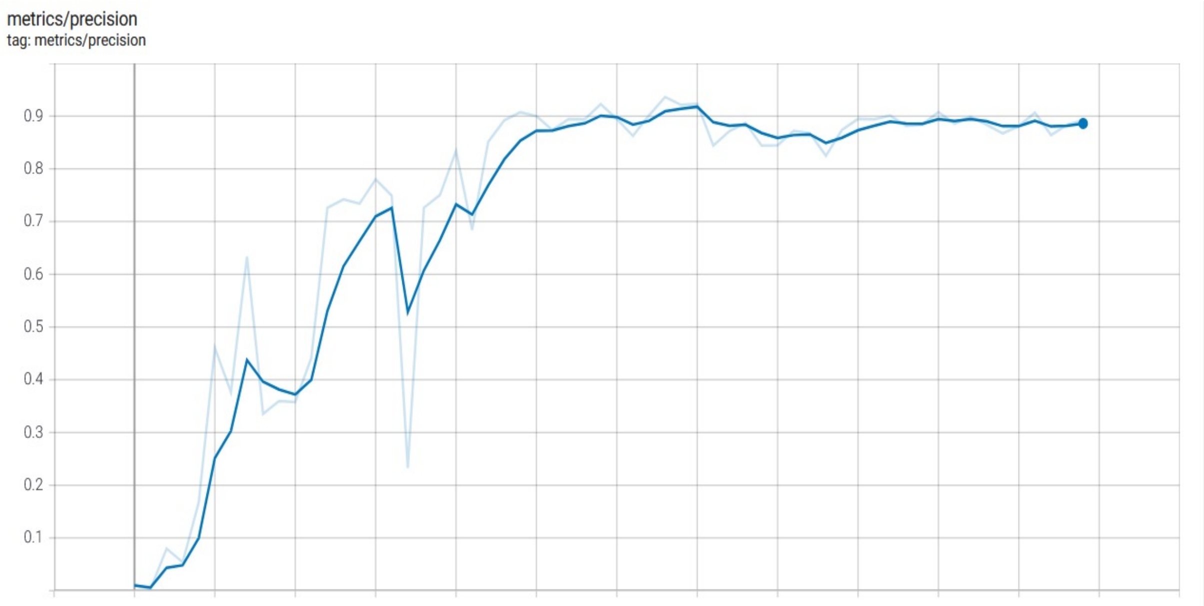


Figure 6.4: Precision

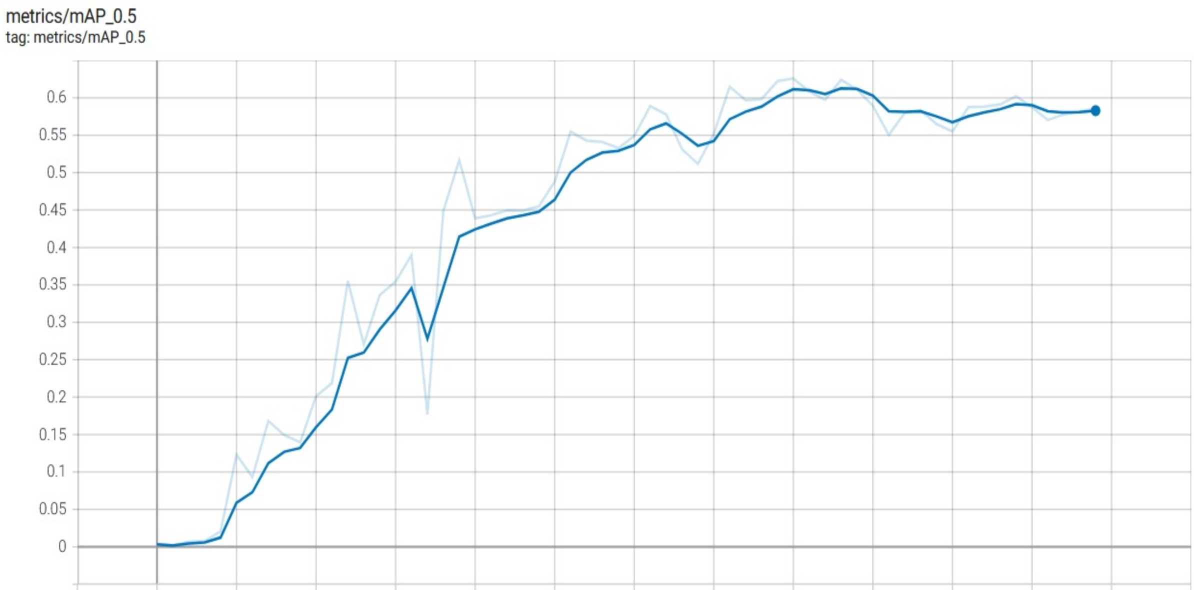


Figure 6.5: Mean Average Precision

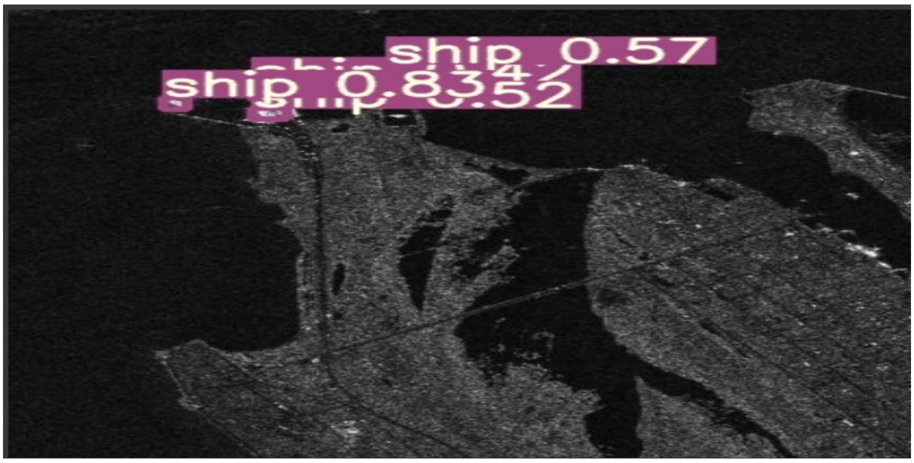
6.4 Location and Size

After the ships have been identified in a given image, a text file of the same name is created that contains the X and Y coordinates of one point of the ship, length, breadth, size (in pixels), longitude and latitude. The size and location were calculated from the bounding box details received from the model.

```
0 0.969952 0.810096 0.0168269 0.0192308
0 0.721154 0.575721 0.0144231 0.0600962
0 0.508413 0.111779 0.0264423 0.0552885
0 0.579327 0.959135 0.0432692 0.0528846
0 0.0288462 0.19351 0.0576923 0.0600962
0 0.371394 0.390625 0.0649038 0.0552885
0 0.127404 0.677885 0.0336538 0.0673077

Size : 56.00001279987712
Location : 11.987442777600002,8.351441683199997

Size : 150.00035520018432
Location : 8.912598055200002,5.935222933199999

Size : 253.0001023999488
Location : 6.2833745844000015,1.1523520667999998

Size : 395.99960320008194
Location : 7.1597865276000014,9.887914541999997

Size : 600.0003807999386
Location : 0.35650441656000004,1.9949332919999994

Size : 620.9999903996927
Location : 4.589984167200001,4.0270312499999985

Size : 391.99950719993853
Location : 1.5745605552000002,6.988452041999998
```

Figure 6.6: Ship Details per Image

6.5 Examples




ship 0.91
ship 0.87


ship 0.85
ship
ship 0.89

CHAPTER 7
CONCLUSION AND FUTURE WORKS

7.1 Conclusion

This YOLO based ship detection system can be used for different purposes like disaster management, military use, port authorities to locate lost ships, surveillance, etc. This system can be used in different weather conditions like during fog, rain, storm, cyclone, etc. due to the benefits of SAR imaging. The system is both fast and accurate so it can also be used for real time monitoring.

7.2 Future Works

In the future, we can try to modify the system to not only get the size and location of the ship but also to compare various images spanning across a given timeline to track the route of any given ship. This can help in figuring out the cause of certain shipwrecks and missing ships. It is also possible to calculate the speed of the ship from the wake.

REFERENCES

[1] S. Arivazhagan, W. S. Lilly Jebarani, R. Newlin Shebiah, S. V. Ligi, P. V. Hareesh Kumar, and K. Anilkumar. Significance based ship detection from sar imagery. In *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pages 1–5, 2019.

[2] B. Bhadran and J. J. Nair. Classification of patterns on high resolution sar images. In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pages 784–792, 2015.

[3] H. Cui, Y. Yang, M. Liu, T. Shi, and Q. Qi. Ship detection: An improved yolov3 method. In *OCEANS 2019 - Marseille*, pages 1–4, 2019.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[5] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[6] H. Tanveer, T. Balz, and B. Mohamdi. Using convolutional neural network (cnn) approach for ship detection in sentinel-1 sar imagery. In *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, pages 1–5, 2019.

[7] Shunjun Wei, Xiangfeng Zeng, Qizhe Qu, Mou Wang, Hao Su, and Jun Shi. Hrsid: A high-resolution sar images dataset for ship detection and instance segmentation. *Ieee Access*, 8:120234–120254, 2020.

[8] T. ZHANG, X. ZHANG, J. SHI, and S. WEI. High-speed ship detection in sar images by improved yolov3. In *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pages 149–152, 2019.

Appendix A
Result Graph

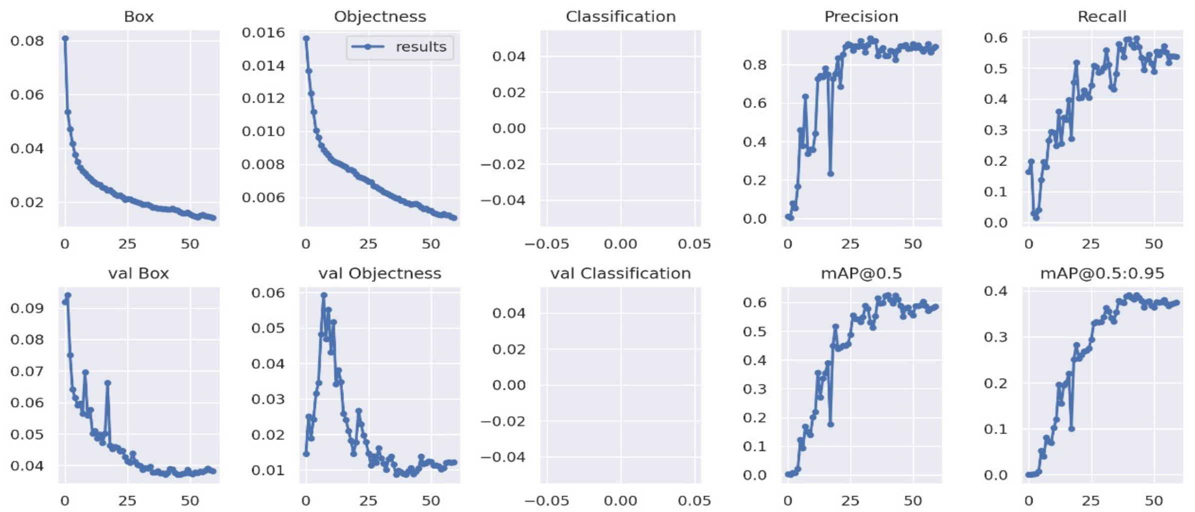
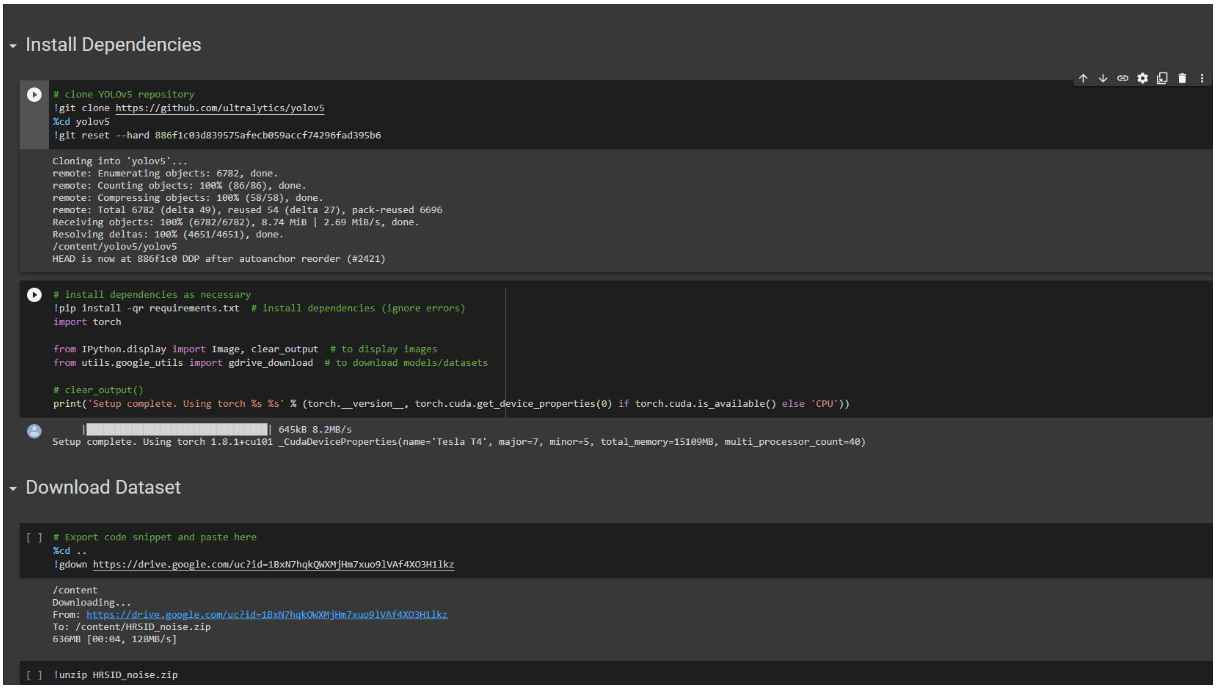
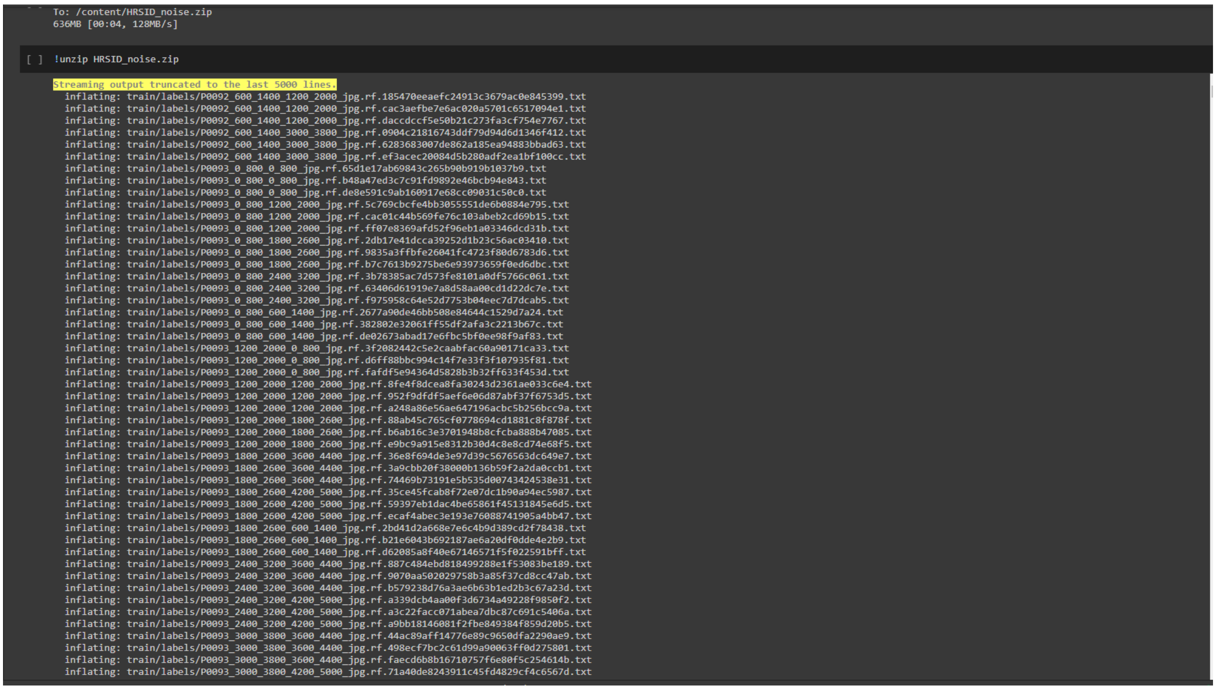


Figure A.1: Precision, Recall and mAP Graph

Appendix B
Code Screenshot





```
inflating: train/labels/P0093_600_1400_2400_3200_jpg.rf.65334be6a2ae10c4564090fa809e6e43.txt
```

```
%cat data.yaml
```

```
train: ../train/images
val: ../valid/images

nc: 1
names: ['ship']
```

Define Model Configuration and Architecture

```
# define number of classes based on YAML
import yaml
with open("data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

```
#this is the model configuration we will use
%cat /content/yolov5/models/yolov5s.yaml
```

```
from IPython.core.magic import register_line_cell_magic

@register_line_cell_magic
def writetemplate(line, cell):
    with open(line, 'w') as f:
        f.write(cell.format(**globals()))
```

```
%%writetemplate /content/yolov5/models/custom_yolov5s.yaml

# parameters
nc: {num_classes}  # number of classes
depth_multiple: 0.32  # model depth multiple
width_multiple: 0.24  # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23]
  - [30,61, 62,45, 59,119]
  - [116,90, 156,198, 373,326]

# YOLOv5 backbone
backbone:
```

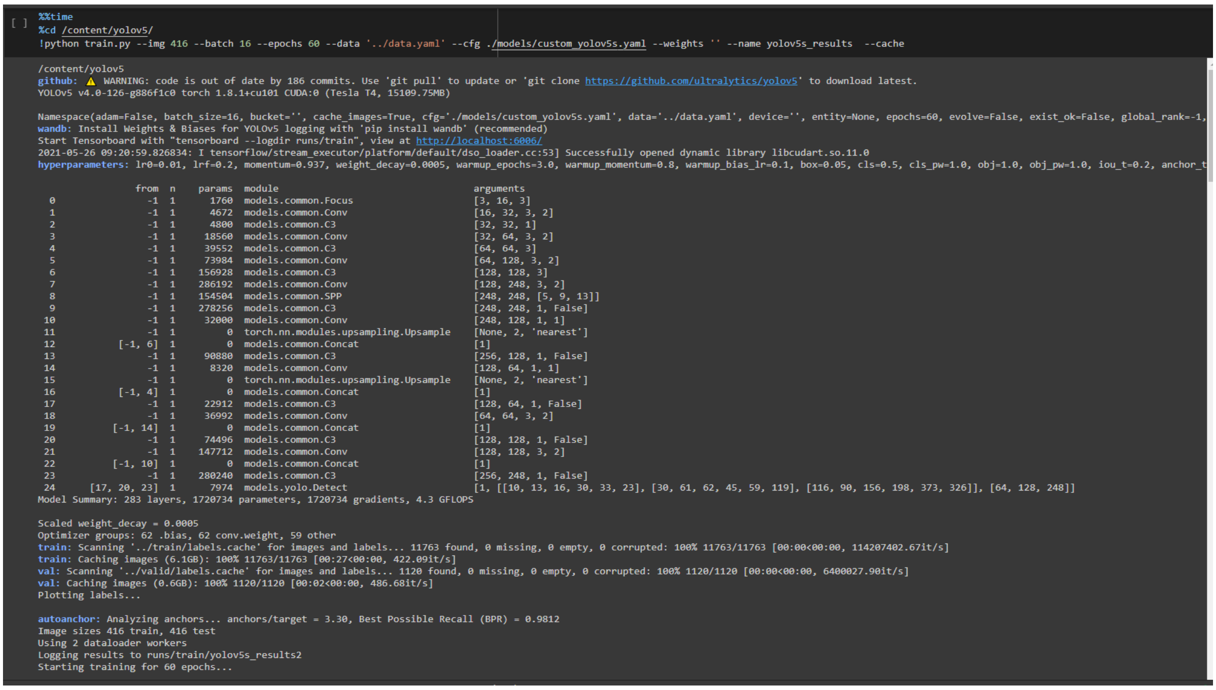
```
   [-1, 1, Conv, [128, 3, 2]],
   [-1, 1, Conv, [256, 3, 2]],
   [-1, 9, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, C3, [1024, False]],
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],
   [-1, 3, C3, [512, False]],

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],
   [-1, 3, C3, [256, False]],

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],
   [-1, 3, C3, [512, False]],

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],
   [-1, 3, C3, [1024, False]],

   [[17, 20, 23], 1, Detect, [nc, anchors]],
  ]
```

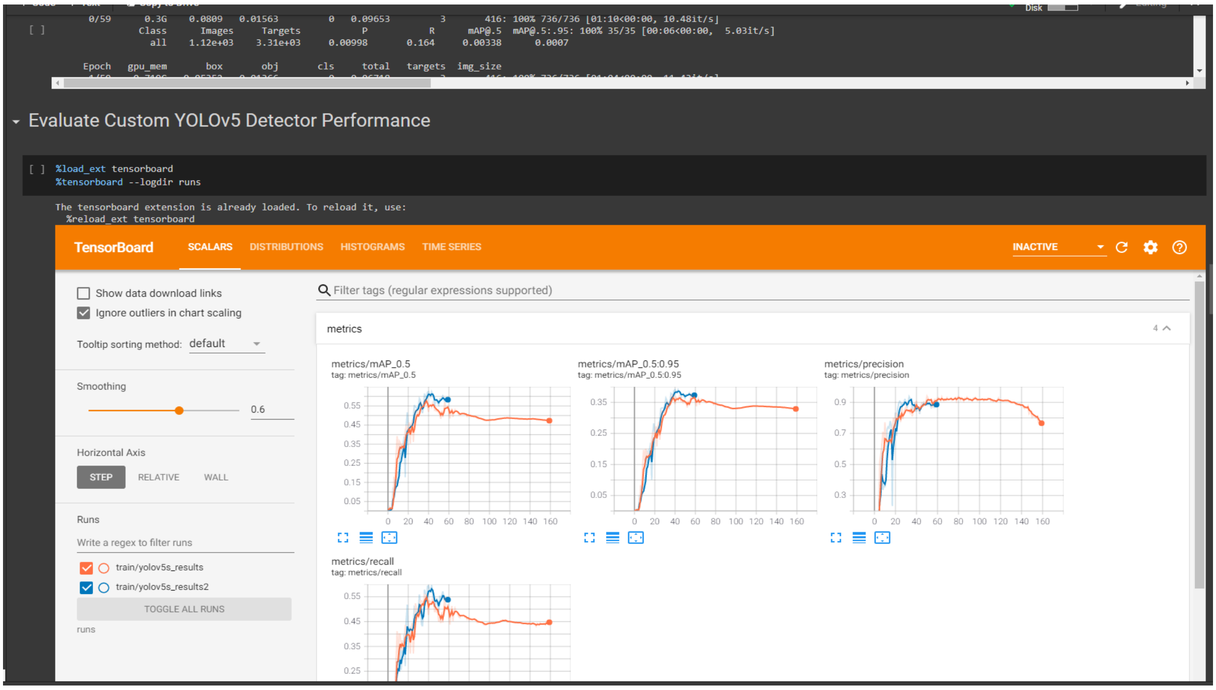
Train Custom YOLOv5 Detector

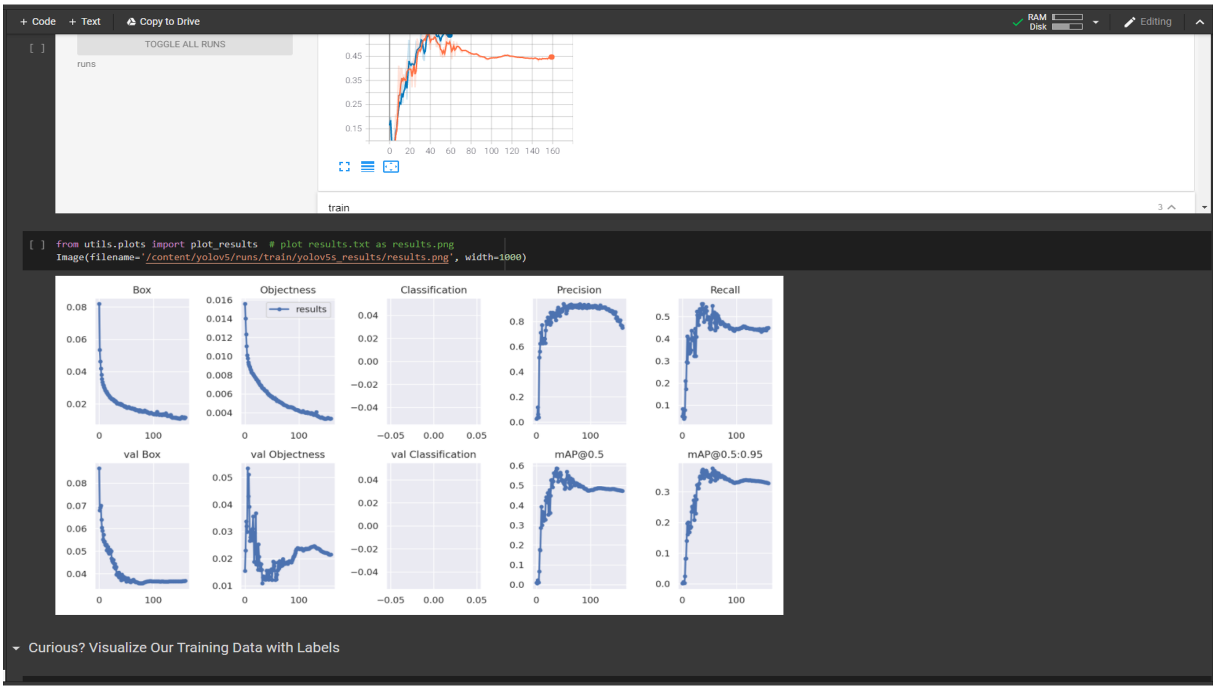
Next, we'll fire off training!

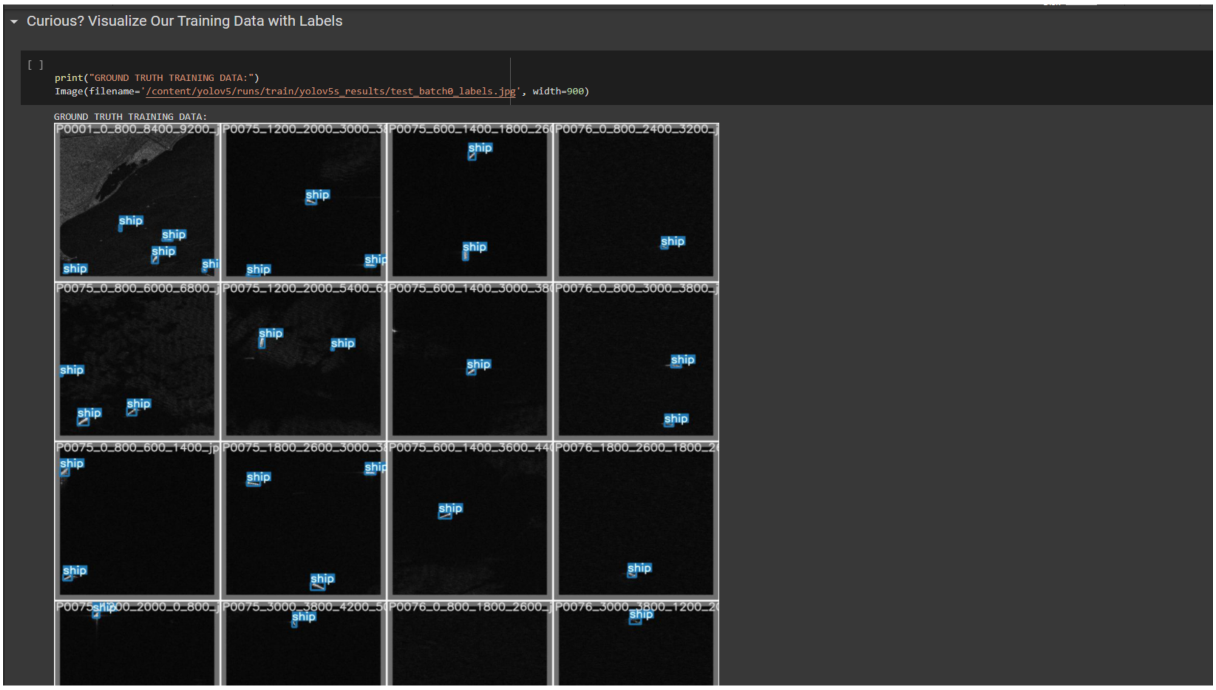
```
# train yolov5s on custom data for 60 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 60 --data '../data.yaml' --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results  --cache
```

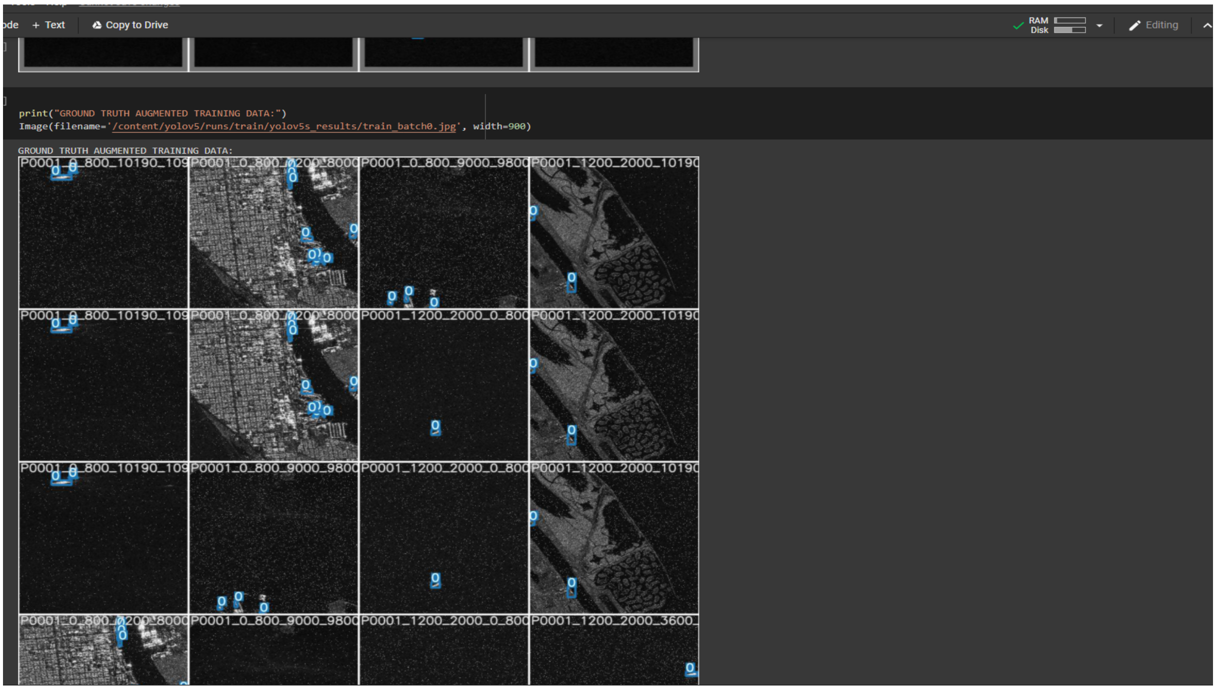
```
/content/yolov5
github: ⚠ WARNING: code is out of date by 186 commits. Use 'git pull' to update or 'git clone https://github.com/ultralytics/yolov5' to download latest.
```

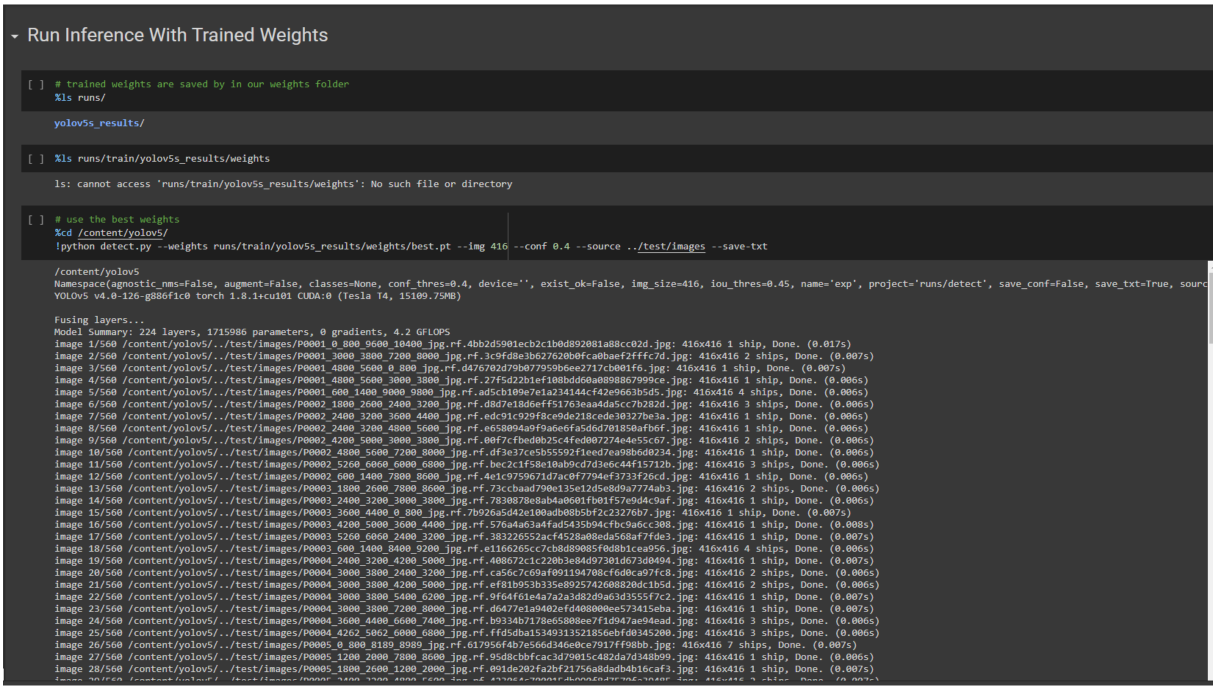

[ ]
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 60 --data '../data.yaml' --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results --cache
/content/yolov5
github: WARNING: code is out of date by 186 commits. Use 'git pull' to update or 'git clone https://github.com/ultralytics/yolov5' to download latest.
YOLOv5 v4.0-126-g886f1c0 torch 1.8.1+cu101 CUDA:0 (Tesla T4, 15109.75MB)
Namespace(adam=False, batch_size=16, bucket='', cache_images=True, cfg='./models/custom_yolov5s.yaml', data='../data.yaml', device='', entity=None, epochs=60, evolve=False, exist_ok=False, global_rank=-1,
wandb: Install Weights & Biases for YOLOv5 logging with 'pip install wandb' (recommended)
Start Tensorboard with "tensorboard --logdir runs/train", view at http://localhost:6006/
2021-05-26 09:20:59.826834: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
hyperparameters: lr0=0.01, lrf=0.2, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=0.05, cls=0.5, cls_pw=1.0, obj=1.0, obj_pw=1.0, iou_t=0.2, anchor_t
from n params module arguments
0 -1 1 1760 models.common.Focus [3, 16, 3]
1 -1 1 4672 models.common.Conv [16, 32, 3, 2]
2 -1 1 4800 models.common.C3 [32, 32, 1]
3 -1 1 18560 models.common.Conv [32, 64, 3, 2]
4 -1 1 39552 models.common.C3 [64, 64, 3]
5 -1 1 73984 models.common.Conv [64, 128, 3, 2]
6 -1 1 156928 models.common.C3 [128, 128, 3]
7 -1 1 286192 models.common.Conv [128, 248, 3, 2]
8 -1 1 154504 models.common.SPP [248, 248, [5, 9, 13]]
9 -1 1 278256 models.common.C3 [248, 248, 1, False]
10 -1 1 32000 models.common.Conv [248, 128, 1, 1]
11 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
12 [-1, 6] 1 0 models.common.Concat [1]
13 -1 1 90880 models.common.C3 [256, 128, 1, False]
14 -1 1 8320 models.common.Conv [128, 64, 1, 1]
15 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
16 [-1, 4] 1 0 models.common.Concat [1]
17 -1 1 22912 models.common.C3 [128, 64, 1, False]
18 -1 1 36992 models.common.Conv [64, 64, 3, 2]
19 [-1, 14] 1 0 models.common.Concat [1]
20 -1 1 74496 models.common.C3 [128, 128, 1, False]
21 -1 1 147712 models.common.Conv [128, 128, 3, 2]
22 [-1, 10] 1 0 models.common.Concat [1]
23 -1 1 280240 models.common.C3 [256, 248, 1, False]
24 [17, 20, 23] 1 7974 models.yolo.Detect [1, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [64, 128, 248]]
Model Summary: 283 layers, 1720734 parameters, 1720734 gradients, 4.3 GFLOPS
Scaled weight_decay = 0.0005
Optimizer groups: 62 .bias, 62 conv.weight, 59 other
train: Scanning '../train/labels.cache' for images and labels... 11763 found, 0 missing, 0 empty, 0 corrupted: 100% 11763/11763 [00:00<00:00, 114207402.67it/s]
train: Caching images (6.1GB): 100% 11763/11763 [00:27<00:00, 422.09it/s]
val: Scanning '../valid/labels.cache' for images and labels... 1120 found, 0 missing, 0 empty, 0 corrupted: 100% 1120/1120 [00:00<00:00, 6400027.90it/s]
val: Caching images (0.6GB): 100% 1120/1120 [00:02<00:00, 486.68it/s]
Plotting labels...
autoanchor: Analyzing anchors... anchors/target = 3.30, Best Possible Recall (BPR) = 0.9812
Image sizes 416 train, 416 test
Using 2 dataloader workers
Logging results to runs/train/yolov5s_results2
Starting training for 60 epochs...


0/59 0.3G 0.0809 0.01563 0 0.09653 3 416: 100% 736/736 [01:10<00:00, 10.48it/s]
Class Images Targets P R mAP@.5 mAP@.5:.95: 100% 35/35 [00:06<00:00, 5.03it/s]
all 1.12e+03 3.31e+03 0.00998 0.164 0.00338 0.0007
Epoch gpu_mem box obj cls total targets img_size
Evaluate Custom YOLOv5 Detector Performance
[ ]
%load_ext tensorboard
%tensorboard --logdir runs
The tensorboard extension is already loaded. To reload it, use:
%reload_ext tensorboard
TensorBoard
SCALARS
DISTRIBUTIONS
HISTOGRAMS
TIME SERIES
INACTIVE
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
train/yolov5s_results
train/yolov5s_results2
TOGGLE ALL RUNS
runs
Filter tags (regular expressions supported)
metrics
4
metrics/mAP_0.5
tag: metrics/mAP_0.5
metrics/mAP_0.5:0.95
tag: metrics/mAP_0.5:0.95
metrics/precision
tag: metrics/precision
metrics/recall
tag: metrics/recall
0 20 40 60 80 100 120 140 160



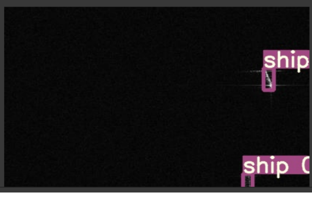



ode + Text Copy to Drive
RAM
Disk
Editing
print("GROUND TRUTH AUGMENTED TRAINING DATA:")
Image(filename='/content/yolov5/runs/train/yolov5s_results/train_batch0.jpg', width=900)
GROUND TRUTH AUGMENTED TRAINING DATA:
P0001_0_800_10190_109
P0001_0_800_7200_8000
P0001_0_800_9000_9800
P0001_1200_2000_10190
P0001_1200_2000_0_800
P0001_1200_2000_3600_


Run Inference With Trained Weights
[ ] # trained weights are saved by in our weights folder
%ls runs/
yolov5s_results/
[ ] %ls runs/train/yolov5s_results/weights
ls: cannot access 'runs/train/yolov5s_results/weights': No such file or directory
[ ] # use the best weights
%cd /content/yolov5/
!python detect.py --weights runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.4 --source ../test/images --save-txt
/content/yolov5
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.4, device='', exist_ok=False, img_size=416, iou_thres=0.45, name='exp', project='runs/detect', save_conf=False, save_txt=True, sourc
YOLOv5 v4.0-126-g886f1c0 torch 1.8.1+cu101 CUDA:0 (Tesla T4, 15109.75MB)
Fusing layers...
Model Summary: 224 layers, 1715986 parameters, 0 gradients, 4.2 GFLOPS
image 1/560 /content/yolov5/../test/images/P0001_0_800_9600_10400_jpg.rf.4bb2d5901ecb2c1b0d892081a88cc02d.jpg: 416x416 1 ship, Done. (0.017s)
image 2/560 /content/yolov5/../test/images/P0001_3000_3800_7200_8000_jpg.rf.3c9fd8e3b627620b0fca0baef2fffc7d.jpg: 416x416 2 ships, Done. (0.007s)
image 3/560 /content/yolov5/../test/images/P0001_4800_5600_0_800_jpg.rf.d476702d79b077959b6ee2717cb001f6.jpg: 416x416 1 ship, Done. (0.007s)
image 4/560 /content/yolov5/../test/images/P0001_4800_5600_3000_3800_jpg.rf.27f5d22b1ef108bdd60a0898867999ce.jpg: 416x416 1 ship, Done. (0.006s)
image 5/560 /content/yolov5/../test/images/P0001_600_1400_9000_9800_jpg.rf.ad5cb109e7e1a234144cf42e9663b5d5.jpg: 416x416 4 ships, Done. (0.006s)
image 6/560 /content/yolov5/../test/images/P0002_1800_2600_2400_3200_jpg.rf.d8d7e18d6eff51763eaa4da5cc7b282d.jpg: 416x416 3 ships, Done. (0.006s)
image 7/560 /content/yolov5/../test/images/P0002_2400_3200_3600_4400_jpg.rf.edc91c929f8ce9de218cede30327be3a.jpg: 416x416 1 ship, Done. (0.006s)
image 8/560 /content/yolov5/../test/images/P0002_2400_3200_4800_5600_jpg.rf.e658094a9f9a6e6fa5d6d701850afb6f.jpg: 416x416 1 ship, Done. (0.006s)
image 9/560 /content/yolov5/../test/images/P0002_4200_5000_3000_3800_jpg.rf.00f7cfbed0b25c4fed007274e4e55c67.jpg: 416x416 2 ships, Done. (0.006s)
image 10/560 /content/yolov5/../test/images/P0002_4800_5600_7200_8000_jpg.rf.df3e37ce5b55592f1eed7ea98b6d0234.jpg: 416x416 1 ship, Done. (0.006s)
image 11/560 /content/yolov5/../test/images/P0002_5260_6060_6000_6800_jpg.rf.bec2c1f58e10ab9cd7d3e6c44f15712b.jpg: 416x416 3 ships, Done. (0.006s)
image 12/560 /content/yolov5/../test/images/P0002_600_1400_7800_8600_jpg.rf.4e1c9759671d7ac0f7794ef3733f26cd.jpg: 416x416 1 ship, Done. (0.006s)
image 13/560 /content/yolov5/../test/images/P0003_1800_2600_7800_8600_jpg.rf.73ccbaad790e135e12d5e8d9a7774ab3.jpg: 416x416 2 ships, Done. (0.006s)
image 14/560 /content/yolov5/../test/images/P0003_2400_3200_3000_3800_jpg.rf.7830878e8ab4a0601fb01f57e9d4c9af.jpg: 416x416 1 ship, Done. (0.006s)
image 15/560 /content/yolov5/../test/images/P0003_3600_4400_0_800_jpg.rf.7b926a5d42e100adb08b5bf2c23276b7.jpg: 416x416 1 ship, Done. (0.007s)
image 16/560 /content/yolov5/../test/images/P0003_4200_5000_3600_4400_jpg.rf.576a4a63a4fad5435b94cfbc9a6cc308.jpg: 416x416 1 ship, Done. (0.008s)
image 17/560 /content/yolov5/../test/images/P0003_5260_6060_2400_3200_jpg.rf.383226552acf4528a08eda568af7fde3.jpg: 416x416 1 ship, Done. (0.007s)
image 18/560 /content/yolov5/../test/images/P0003_600_1400_8400_9200_jpg.rf.e1166265cc7cb8d89085f0d8b1cea956.jpg: 416x416 4 ships, Done. (0.006s)
image 19/560 /content/yolov5/../test/images/P0004_2400_3200_4200_5000_jpg.rf.408672c1c220b3e84d97301d673d0494.jpg: 416x416 1 ship, Done. (0.007s)
image 20/560 /content/yolov5/../test/images/P0004_3000_3800_2400_3200_jpg.rf.ca56c7c69af091194708cf6d0ca97fc8.jpg: 416x416 2 ships, Done. (0.006s)
image 21/560 /content/yolov5/../test/images/P0004_3000_3800_4200_5000_jpg.rf.ef81b953b335e8925742608820dc1b5d.jpg: 416x416 2 ships, Done. (0.006s)
image 22/560 /content/yolov5/../test/images/P0004_3000_3800_5400_6200_jpg.rf.9f64f61e4a7a2a3d82d9a63d3555f7c2.jpg: 416x416 1 ship, Done. (0.007s)
image 23/560 /content/yolov5/../test/images/P0004_3000_3800_7200_8000_jpg.rf.d6477e1a9402efd408000ee573415eba.jpg: 416x416 1 ship, Done. (0.007s)
image 24/560 /content/yolov5/../test/images/P0004_3600_4400_6600_7400_jpg.rf.b9334b7178e65808ee7f1d947ae94ead.jpg: 416x416 3 ships, Done. (0.006s)
image 25/560 /content/yolov5/../test/images/P0004_4262_5062_6000_6800_jpg.rf.ffd5dba15349313521856ebfd0345200.jpg: 416x416 3 ships, Done. (0.006s)
image 26/560 /content/yolov5/../test/images/P0005_0_800_8189_8989_jpg.rf.617956f4b7e566d346e0ce7917ff98bb.jpg: 416x416 7 ships, Done. (0.007s)
image 27/560 /content/yolov5/../test/images/P0005_1200_2000_7800_8600_jpg.rf.95d8cbbfcac3d79015c482da7d348b99.jpg: 416x416 1 ship, Done. (0.006s)
image 28/560 /content/yolov5/../test/images/P0005_1800_2600_1200_2000_jpg.rf.091de202fa2bf21756a8dadb4b16caf3.jpg: 416x416 1 ship, Done. (0.007s)

```
image 50/560 /content/yolov5/../test/images/P0009_4200_5000_6000_6800_jpg.rf.292de049751593ae477b427fcae58652.jpg: 416x416 1 ship, Done. (0.007s)
image 51/560 /content/yolov5/../test/images/P0009_4260_5060_2400_3200_jpg.rf.4a14492003c6467eba72a982dd3ba676.jpg: 416x416 1 ship, Done. (0.007s)
image 52/560 /content/yolov5/../test/images/P0010_0_800_8400_9200_jpg.rf.4dd998d6660601e2d1ee736a3a64ca71.jpg: 416x416 4 ships, Done. (0.006s)
```

```
#display inference on ALL test images
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("\n")
```




```
import os
def calc_size(w,h):
    w1 = w*416
    h1= h*416
    size = w1*h1
    return size
def calc_location(x,y):
    #Lattitude and Longitude
    BL_lat = 46.6758
    BL_long= 75.8976
    TR_lat = 56.9850
    TR_long= 88.2564
    long_diff = TR_long-BL_long
    lat_diff = TR_lat -BL_lat
    x1=x*long_diff
    y1=y*lat_diff
    return x1,y1
def size_and_location(file_name):
    fp=open(file_name,"r")
    i=0
    s=[]
    y1=[]
    for line in fp:
        ln=line.split(" ")
        x=float(ln[1])
        y=float(ln[2])
        w=float(ln[3])
        h=float(ln[4])
        s.append( calc_size(w,h))
        x1.append(0)
        y1.append(0)
        x1[i],y1[i] = calc_location(x,y)
        i+=1
    fp.close()

    fp=open(file_name,"a")
    fp.write("\n")
    for i in range(len(s)):
        fp.write("\n")
        fp.write("Size : "+str(s[i])+"\n")
        fp.write("Location : "+str(x1[i])+","+str(y1[i])+"\n")
    fp.close()

file_names=os.listdir()
print (file_names)
for file_name in file_names:
    size_and_location(file_name)
```