National College of Ireland

Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) of Ships using Data Augmentation and Deep Learning

MSc Research Project
MSc in Data Analytics

Devashish Vijay Rayate
Student ID: X19232616

School of Computing
National College of Ireland

Supervisor: Prof. Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Devashish Vijay Rayate
Student ID:	X19232616
Programme:	MSc in Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Prof. Hicham Rifai
Submission Due Date:	16/08/2021
Project Title:	**Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) of Ships using Data Augmentation and Deep Learning**
Word Count:	6464
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Devashish Vijay Rayate
Date:	19th September 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) of Ships using Data Augmentation and Deep Learning

Devashish Vijay Rayate
X19232616

Abstract

Automatic Target Recognition (ATR) using Synthetic Aperture Radar (SAR) has become a popular issue in studies, and it's crucial for water target monitoring. Because SAR imagery are difficult to interpret directly, different machine learning methodologies have been implemented in recent times to recognize maritime objects from SAR data. Since standard deep learning models could only go so far, their effectiveness is limited, and they require more time to train. They often face overfitting issue owing to the unavailability of enough satellite imagery. To solve the SAR-ATR challenge of detecting maritime objects, this paper proposes a deep learning model called Mask R-CNN which uses ResNet101 as a backbone. The model can successfully recognize and segment ships. Data augmentation procedures such as rotating, flip, contrast and brightness modification, fading and intensifying the image are used to overcome challenges with limited SAR images. On the grounds of mean average precision (mAP), the model was analyzed with and without data augmentation. According to the results, the model's mAP improved from 48.9% to 71.52% when it was trained using data augmentation and hyperparameter optimization.

Keywords- SAR, deep learning, convolution neural network, mask R-CNN, ResNet-101, data augmentation, hyperparameter optimization.

1 Introduction

SAR (Synthetic Aperture Radar) is a satellite imaging methodology that provides high-resolution photographs without the use of sunlight or ideal weather conditions. Optical imaging and SAR imaging are not the same. Because optical sensors rely on light from external sources, like the sun, while a radar equipment sends out microwave signals and then analyzes the **reflection** of received signals. Because of its high-resolution photographs producing capabilities in any weather conditions SAR system is being utilized to survey territory where humans have been unable to enter for years. SAR is used in a variety of military and industrial setups. In defense industry, SAR imaging is used for arms targeting systems, battleground intelligence gathering, combat motor vehicle and warship tracking, and surveillance programs. In industrial setups, SAR imaging is used for mining and geomorphology, iceberg monitoring, oil pollution detection, meteorology, flood forecasting and rescue operations, and a variety of others. SAR-ATR (Automatic Target Recognition) (Yang et al.; 2019) is an important application in the computer vision field that involves the detection of targets using SAR imagery.

In past years, the detection of ship targets in SAR data is becoming a popular area in study. The effectiveness of the conventional automatic target recognition (ATR) algorithm rapidly degrades as the SAR image resolution rises. Deep networks have emerged as a novel approach for SAR imagery ship identification (Hou et al.; 2020). Ship identification is one of the military and industrial applications in SAR image analysis, and the aim of this research is on developing it. The requirement for this solution arose since keeping an eye on what's happening on the water is not easy and expensive. With the use of SAR-ATR for maritime surveillance, a nation can be benefited in many ways as the water bodies will be monitored 24 × 7 and any suspicious activity would be caught at early stage. SAR visuals are more difficult to comprehend than optical images since they only provide the amplitude details of the scattered center, causing them difficult to analyze instantly. As a result, target recognition from SAR images has become quite a major topic of study (Xue and Bai; 2019). Hence image analysis algorithms were created to assist. As technological advances continue, several researchers conceived and constructed automated algorithms to locate objects from SAR imagery, and deep learning came into picture. As detailed in the Related Literature, many recognition algorithms for recognizing marine objects have been presented, including shadow compensation, sparse coding, deep learning, hybrid learning, luminance analysis, and many more. Algorithms based on deep learning have considerably improved target identification. Following AlexNet's breakthrough in 2012, new deep learning models were designed in following years to tackle the SAR target recognition challenge (Zhu, Lin, Leung, Leung and Theodoidis; 2020). GoogLeNet, AlexNet, GAN, Graph CNN, ResNet-18, Faster R-CNN, and several other artificial neural networks were applied on different datasets containing SAR marine targets. But, due to the possibility of overfitting, most networks may not go very deep, therefore this study presents a novel technique of creating a Mask R-CNN based SAR-ATR system with ResNet-101 as the backbone, which can go significantly deeper without hurting the model's overall quality.Overfitting is a typical problem with Convolutional Neural Networks, which could be induced by a shortage of training data, since SAR training set is not widely available. Therefore, researchers came up with the technique of data augmentation, which enables a little amount of data to be substantially increased by simple graphic transformations.

1.1 Research Question and Objective

To what extent modern deep learning technique Mask R-CNN combined with data augmentation assist in the recognition of marine vehicles from SAR images?

The fundamental aim of this research is to use the Airbus Intelligence dataset of commercial ships to implement the instance segmentation oriented deep learning technique Mask R-CNN. A secondary aim of the study is to investigate how data augmentation will improve the performance of Mask R-CNN model. The model will be trained with different combinations of hyperparameters and its performance will be assessed using different evaluation metrics.

This paper is divided into following sections:

Section 2: Encompasses discussion of the related studies conducted in SAR imaging.
Section 3: Describes proposed research methodology.
Section 4: Contains design specifications.
Section 5: Presents the implementation of model.
Section 6: Illustrates the evaluation of model.
Section 7: Conclusion and future scope of this research is presented.

2 Related Work

2.1 SAR Object Detection Using Deep Learning

In the field of SAR imagery, ship recognition is a major topic of study. In order to identify ships in SAR data, experts have implemented a number of deep learning models. Standard deep networks, on the flip side, are constricted in terms of depth and training performance. To counter this issue, the SAR ship identification system was built with a very deep CNN ResNet having rapid learning rate and greater performance (Li et al.; 2019). ResNet is different from other CNNs because, it learns on residuals instead of feature vectors. Residual training is easier than direct learning because the outcome does not diminish when the residual is 0. A much more deep neural network can be trained with the residual learning approach. To recognize ships in SAR imagery, (Li et al.; 2019) employed a ResNet50 deep convolutional neural network with 50 layers. Conv1_x - conv4_x were used to extract the features of a SAR ship imagery and produce feature maps. To establish regions of interest (RoIs) which might contain targets, a region proposal network (RPN) is employed. The resulting scale was determined using RoI pooling. Finally, conv5_x was used in bbox regression and categorization. The classifier was tested on a custom constructed dataset of 2900 SAR pictures comprising 7524 boats, and its performance was assessed to that of the YOLOv2 and VGG16 architectures. ResNet50 spent 11 hours to train and attained 94.7% recognition rate, whereas YOLOv2 and VGG16 spent 9.5 and 23 hours, respectively, for training and managed to attain 87.4% and 93.2% accuracy.

The neural network model is a particularly interesting sort of supervised learning (Fu et al.; 2018). Generally CNN models tend to overfit due to the limited samples, making it challenging to attain excellent recognition results. A denser network may theoretically offer higher learning outputs, but it often tends to make the learning process more challenging, especially when sample data is small (Fu et al.; 2018). Several investigations have demonstrated that residual learning could help to eliminate this problem. To solve the SAR-ATR problem, a deep residual network (ResNet-18) was developed, and dropout layers were introduced into building blocks to avoid overfitting caused by inadequate SAR inputs (Fu et al.; 2018). The ResNet-18 design has 18 layers, and certain changes were made to accommodate the MSTAR dataset. The initial layer is accompanied by 16 3×3 convolutional layers to decrease noise present and boost the volume of channels (Fu et al.; 2018). A dropout layer was introduced across two adjacent convolutional layer, which will also occasionally exclude a convolution layer throughout the learning phase to prevent overfitting (Fu et al.; 2018). The network was trained using this strategy, and results found that it enabled the algorithm attain 99.67 percent accuracy, which is 4.04 percent higher than that of the ResNet benchmark (Fu et al.; 2018). Therefore, with a short dataset, the ResNet-18 design is efficient in addressing the SAR-ATR issue.

Deep learning-based ship target identification in SAR imagery is subdivided into discovery and characterization, but they're not combined yet. (Hou et al.; 2020) proposed a deep network-based integration solution for ship target identification and classification based on the process flow of a typical ATR system. The suggested method uses RetinaNet as the underlying network, with the squeeze-and-excitation (SE) subsystem introduced to the categorization subnet at the later part. The focus loss is developed to handle the one-stage target identification case wherein the foreground and background classes are extremely imbalanced during training. Focus loss can substantially minimize the impact of water clutter and mainland false alarm targets in SAR data, making it extremely important for target identification. The SE module consists of two fully connected layers, one pooling layer, and a gaussian polynomial activation layer. The SE configuration was implemented at the end of the classifying subnetwork's second convolution. The approach was not only decreased false alarms but also delivered a significant identification accuracy, according to the results of the testings. (Kun and Yan; 2020) proposed an enhanced YOLOv4-Tiny recognition technique. The focus mechanism unit was added to the modified algorithm to increase extraction of features and try making the object characteristics more noticeable. The Batch normalization optimizer data set was utilized to improve the learning model's stability and significantly prevent gradient loss or inflation. To enhance the training rate and accelerate the fitting of deep neural networks, cosine annealing was utilized. The bayesian inference method was applied to improve the performance index in deep network. SSDD was the test data set used in this work. There were 1,160 images and 2,456 boats in all in SS-DD. There were an average of 2.12 boats per picture. Unlike PASCAL VOC, there were over 9000 pictures divided into 20 subcategories. The scales of the network infrastructure were first produced via learning on the VOC data set, and afterwards the total weight values were learned using transfer learning on the SSDD training sample. And they're all part of the same testing dataset. The mAP of the revised Yolov4-Tiny method was 75.56 percent, whereas the FPS was 30 in the experimental studies. When the backdrop situation is challenging, conventional CFAR-based approaches aren't ideal.

Inspired by how humans analyze a picture, (Yang et al.; 2019) proposed the YOLO (You Only Look Once) strategy, which approaches object detection and classification as a regression task and using regression to determine the object's placement and category. The system is based on the improved GoogLeNet framework, which is capable of extracting low-resolution Satellite imagery with intricate compositions (Yang et al.; 2019). This 36-layer network architecture has 24 convolution layers, 4 max-pool layers, and 2 fully connected layers. The system was evaluated on satellite imagery of maritime objects such as boats, and the findings showed that even for the objects with varied sizes and modest background alterations, superior identification results could be obtained. To achieve promising performance, continuous training is needed for many objects and complicated backgrounds (Yang et al.; 2019). Even if the model still mistakenly identified noise or mountains as ships, resulting in false alerts, CNN has demonstrated considerable potential on SAR-ATR and it has obtained outstanding quality as a supervised deep learning model well-suited for computer vision (Zheng et al.; 2019). Nonetheless, SAR imagery are scarce and difficult to come by, and producing labels for input data manually is time-consuming (Zheng et al.; 2019). This can affect the deep learning model as, a large number of labelled data is needed for a deep convolutional neural network. To overcome this, (Zheng et al.; 2019) proposed a semi-supervised recognition method based on a generative adversarial network (GAN) and a convolutional neural network (CNN). In this method, the GAN generates unlabeled photos and delivers them to CNN along with the original labeled images. The proposed system contains a continuously adaptable multi-discriminator GAN (MGAN) structure to resolve the problems of unstable training (Zheng et al.; 2019). Extensive tests on the MSTAR dataset were undertaken to assess the performance of the MGAN-CNN. The results revealed that the MGAN-CNN outperforms previous SAR ATR techniques in terms of CNN classification accuracy, and that it is suitable for SAR image identification with a limited dataset.

CNNs such as AlexNet (Kechagias-Stamatis and Aouf; 2019), GAN (Zheng et al.; 2019), GoogLeNet (Yang et al.; 2019), and ResNet (Fu et al.; 2018) were not implemented until recently to solve SAR-ATR problem. (Kechagias-Stamatis and Aouf; 2019) presented a multi-model data fusion technique for automatic target recognition using satellite imagery. To optimize the AlexNet CNN's capacity to operate from the optical to the SAR field, a layer-clustering technique was first created. To bridge the optical to SAR image processing technology gap, a clustered-CNN was combined with a multi-class SVM classifier (Kechagias-Stamatis and Aouf; 2019). Finally, utilizing a decision level approach that modifies the fusion weights optimally according to scene properties, the Clustered CNN and the adaptive sparse coding structure were creatively combined. The framework was validated using the MSTAR dataset, and study found that it had precision of 99.33 percent and 99.86 percent for 3-class and 10-class ATR problems, respectively. (Wu et al.; 2020), on the other hand, looked at the effects of different preprocessing and picture enhancing methods on the effectiveness of SAR ATR. The merging of CNN and Support vector machine (SVM), a popular traditional machine learning approach, has gained a lot of emphasis in the SAR-ATR field. Because of SVM's ability to efficiently analyze high-dimensional data, AlexNet's design was updated to develop a new architecture appropriate for SAR ATR, which then was combined with SVM. The MSTAR dataset was partitioned into train and test set after being preprocessed with method of translation augmentation. The features extraction capability of AlexNet and the classifying effectiveness of SVM were then evaluated. Experiments with AlexNet demonstrated that applying data pretreatment and optimization methods before to the deep-learning stage are not required (Wu et al.; 2020). The proposed fusion framework was able to accurately solve the 10-class SAR-ATR question.

Neural network based methods have been employed to retrieve target properties of the original SAR picture in Euclidean, which requires a large number of training set and requires a lot of time to train (Zhu, Lin, Leung, Leung and Theodoidis; 2020). To overcome this issue, a new way to target detection from SAR imaging was devised, emphasizing on the targeted pixel grayscale drop using a graph layout termed Graph CNN, which is very distinctive from the typical deep neural networks so far (Zhu, Lin, Leung, Leung and Theodoidis; 2020). In this approach, the complete grayscale range of a single SAR image is split into multiple subbands, and a node is assigned to show each pixel from every subband with the omitted pixel grayscale sequence. After that, the raw SAR image could be converted from Euclidean to graph data structure. Lastly, a graph CNN is designed to extract features from the previously generated graph-structured inputs and categorize the object. The test findings on the MSTAR data demonstrated that the Graph CNN network achieved average recognition accuracy of 100 percent for the first time in the SAR-ATR field, beating all state-of-the-art approaches (Zhu, Lin, Leung, Leung and Theodoidis; 2020).

(Dong et al.; 2019) developed an improved Faster R-CNN framework and SSD structures to solve the SAR-ATR problem. For object identification, SSD is a one-staged CNN with multiple-scale feature matrices formed by several layers. The pictures were resized to 600*725 pixels to avoid losing details due to forced scaling, but keeping the actual width and height proportion (Dong et al.; 2019). To capture features at different levels, SSD builds region proposals at several levels in the network, producing in a pyramid of deep hierarchical vectors (Dong et al.; 2019). The Faster R-CNN architecture is made up of two CNNs: the Regional Proposal Network (RPN) and Fast R-CNN. RPN is composed of two branches and added convolutional layers, whereas Fast R-CNN extracts region proposals from raw data as a decoder. RPN creates proposals for regions of interest, which are subsequently utilised Fast R-CNN to construct bounding box. Faster R-CNN algorithms sharpened slightly in performance on the MSTAR dataset, while SSD networks were operationally effective and significantly faster with outstanding accuracy.

Deep learning-based algorithms, like Faster RCNN, currently hold a strong place in the area of optical target recognition. On the other hand, is ineffective at detecting SAR ship targets. (Gui et al.; 2019) proposed a scale transference subsystem for SAR ship recognition. To obtain parametric characteristics, the scale-transfer function connects to numerous feature maps rather than using an one feature vector. In addition, the RoIAlign technique was employed to estimate the bounding box precision. Background features are added to the recognition subnetwork to aid in the identification of complicated objects. Experiments using the SAR ship detection dataset (SSDD) reveal that the suggested strategy outperforms the conventional models in terms of prediction performance.

The majority of existing algorithms recognize ships by applying a rectangular bounding box, however they do not do pixel-level segmentation. Relying on an enhanced Mask R-CNN framework, (Nie et al.; 2020) proposed a ship recognition and segmentation framework. At the pixel level, the suggested technique was reliably able to identify and segment maritime targets. The link between the top-level layer and the lower-level layers were simplified by introducing FPN, a bottom-up approach to Mask R-CNN's architecture, enabling the lower layer characteristics to be more efficiently exploited at the top layer. In the bottom-up design, they assigned weights to every channel using channel-wise focus and allocated a corresponding value to every pixel in the feature vectors using the spatial attention system. This helps the feature vectors to respond appropriately to the attributes of the object. Experimental results on 'Airbus Ship Detection Challenge' dataset showed that the modified Mask R-CNN model was able to achieve mAP of 76.1%.

2.2 Data Augmentation

Due to its rapid advancement, deep learning has achieved substantial breakthroughs in image analysis technology over the years. Regardless of the notion that machine learning has several advantages and has produced considerable accomplishments, it and its related techniques still confront several difficulties (Guohang et al.; 2020). That is, there are insufficient training examples or an unbalanced class balancing in the dataset (Guohang et al.; 2020). Data augmentation is a method of expanding the amount of a data by modifying existing data or creating new data form previously gathered data (Ding et al.; 2017).

Regular geometrical alterations to image data, such as contrast enhancement, resizing, spin, and translating, are part of traditional data augmentation (Chen and Cao; 2019). For data augmentation, GANs (Generative Adversarial Networks) have lately gained popularity. This framework is made up of two networks: one that generates false visuals and the other which recognizes actual and false data in real time (Chen and Cao; 2019). This strategy was tested using the diabetic retinopathy lesion images dataset.

The CNN framework was trained on three datasets: the initial, one augmented with traditional approaches, and one augmented using GAN. The finding showed that employing a combination of standard data augmentation approach and GAN for healthcare information learning with little data was much more efficient.

Synthetic aperture radar captures the field object with a depression angle, producing in a shadow region in SAR pictures (Zhu, Hon, Wong, Leung, Lin and Lin; 2020). As a result, in this scenario, the shadow readily hides a number of the SAR picture's original elements, thus impacting the SAR automatic object identification study (Zhu, Hon, Wong, Leung, Lin and Lin; 2020). Data augmentation utilizing pixel complement, a completely new approach for restoring the hidden area of the SAR picture's object, was proposed to tackle this problem. Firstly, create a simulated optic target model images with the same SAR capture perspective as the raw SAR image (Zhu, Hon, Wong, Leung, Lin and Lin; 2020). Generate pixel value relationships for the targeted area in both the optical model picture and the actual SAR picture, and then rebuild the hidden section of the ground target in the actual SAR image with pixel value computation based on the image pixel association in the optical model images (Zhu, Hon, Wong, Leung, Lin and Lin; 2020). Studies on the MSTAR dataset demonstrated that this strategy increases target recognition accuracy for several SAR ATR classifiers, with an accuracy rate of 97.45%, which is a significant result for SAR ATR with small dataset (Zhu, Hon, Wong, Leung, Lin and Lin; 2020).

2.3 Literature Summary and Analysis

The above-mentioned studies and research have influenced this research project. In subsection 2.2, implementations of data augmentation were surveyed, whereas in subsection 2.1, implementations of deep learning approaches to tackle the SAR-ATR challenge were outlined. Several deep learning models showed positive outcomes for SAR target detection, however because of a shortage of dataset and data augmentation, analysts have been unable to analyze their complete potential. On the contrary side, deep learning approaches coupled with certain data augmentation strategies were effective, however they takes too much time to train. As a result, we can explore into deep learning approaches that could go extremely deep without adversely effecting the overall performance of the neural net. Mask R-CNN is gaining popularity recently due to its ability to detect and segment images efficiently than other deep learning models(He et al.; 2017). Moreover, because neural networks require a vast amount of data, data augmentation can help to bring value in this research.

3 Research Methodology

All the different deep learning models discussed in literature review section are working at their best however, the most common problem that they are facing is limited size of dataset and smaller neural network designs that doesn't go too deep due to possibility of overfitting. To counter this challenge, Mask R-CNN deep learning model is implemented in this research along with different data augmentation techniques. Mask R-CNN is a straightforward and effective object segmentation approach for applications like human posture identification. The COCO 2016 Competition awarded it first position. Faster R-CNN for object recognition and FCN for image segmentation are combined in Mask R-CNN. FCN is utilized for mask forecasting, boundary processing, and categorization once the Faster R-CNN finds the object. Mask R-CNN is an useful approach for SAR image recognition and segmentation because of the successful combination of the two (He et al.; 2017).

The Cross-industry standard data mining process (CRISP-DM) is used to carry out this research. CRISP-DM is segmented into six parts, as illustrated in the Figure 1 below.

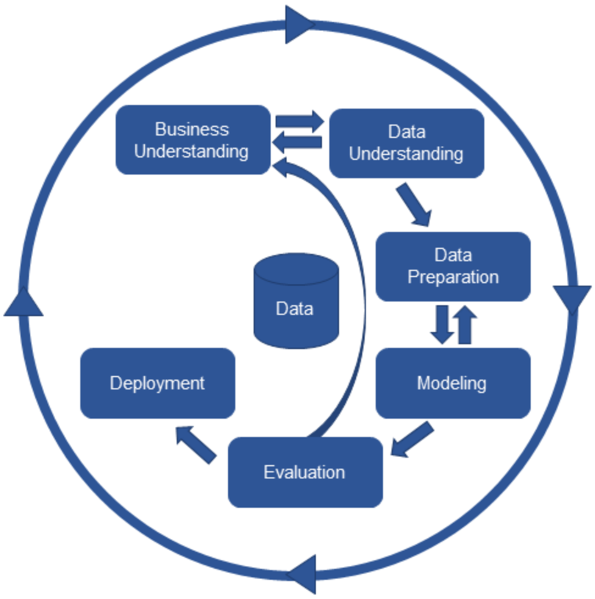


Figure 1: CRISP-DM Model (Vorhies; 2016)

3.1 Business Understanding

Deep neural networks are quite excellent at identifying ships from SAR data in real time. Since physically recognizing these objects is time-consuming, the application of artificial intelligence in the commercial sector could be a crucial factor. Realizing that conventional deep learning models can be unstable at times could allow for more precise and faster results when employing newer deep learning models. This will help officials determine whether any ships are in banned regions or if any ships are stuck in the water due to bad weather, so they can plan how to approach them and block or rescue them. Early identification can aid government officials in preventing illicit activities that included water as medium of transport and rescuing those who may have become stranded in the middle of the water owing to bad weather. The goal of this study is to develop an automated system that can recognize ships and boats of various sizes from SAR imagery.

3.2 Data Understanding

The Airbus Space and Defence group provided the "Airbus ship detection challenge" dataset on Kaggle, which is being used in this project [1]. The dataset includes a variety of SAR imagery of ocean surfaces (as shown in figure below), each with a distinct number

[1] https://www.kaggle.com/c/airbus-ship-detection

of ships and boats. There are also some photographs that are devoid of any ships. An annotation file with the image name and bounding box information is also provided which can be used to train the model. Because all of the files are in JPEG format, they can be used straight away. The proposed Mask R-CNN method is trained, tested, and evaluated using this dataset.

3.3 Data Preparation

The downloaded data is stored in 2 different folders: train and val. The train folder contains SAR images which will be used for training and testing of the model whereas, val folder contains SAR images that will be used for validation. An annotation file is also downloaded with the data which contains bounding box information of ships inside the SAR image in run length encoding format. There is no need to transform the data because the files are already in JPEG format hence the input images can be used right away.

3.4 Data Preprocessing

There are around 12,500 SAR images provided for training and testing. There are some images that do not contain any ships or are corrupted. Those images were filtered out before splitting the data which makes a total image count available for training and testing as 11,332. There are around 5600 images available for final model validation.

3.4.1 Data augmentation

To expand the volume of the dataset, simple data augmentation procedures such as rotation, flipping, brightness modification, contrast adjustment, blurring image, and sharpening image are used. Augmentation of a sample image is shown in Figure 2 below.

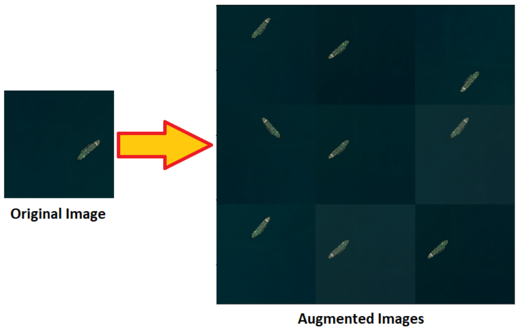


Figure 2: Data Augmentation on SAR image

3.4.2 Splitting of data

The given dataset is split into training and testing dataset with 80:20 ratio respectively.

3.5 Modeling and Evaluation

In this section the modeling of proposed Mask R-CNN model is described. The number labels in the workflow diagram (Figure 3) shows the phases in the modeling of neural network.

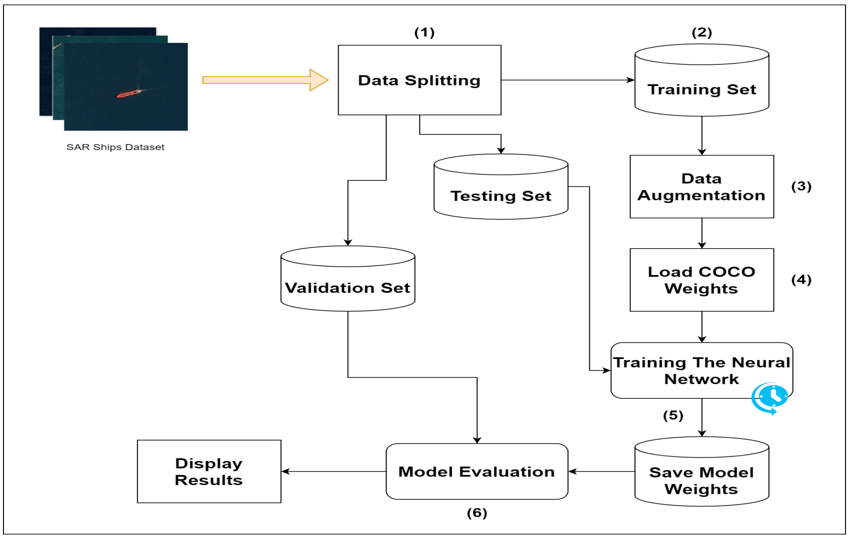


Figure 3: Modeling Workflow Diagram

1. Firstly, the images are loaded into the system and split into train, test and validation datasets.

2. After data splitting, the training set goes through real time data augmentation where it is expanded.

3. In this step, the pre-trained COCO weights are loaded into the model for transfer learning.

4. The model training is done in this step where training set and testing set are used.

5. During every epoch, the generated model weights are stored and at the end of training phase, the epoch with minimum validation loss is selected for model evaluation.

6. At this stage, the model is evaluated using different evaluation metrics and random samples from validation set are visualized.

4 Design Specification

4.1 Mask R-CNN

Mask R-CNN with ResNet-101 as backbone is used in this research to solve the problem of SAR-ATR for maritime vehicles detection. The model is implemented using Tensorflow 1.15.2 and Keras 2.1.5 libraries. The proposed methodology was inspired by reviewed literature (He et al.; 2017). Mask R-CNN is a deep neural network designed to handle the problem of object segmentation in artificial intelligence. To put it another way, it can distinguish between various things in a picture or video. It takes a picture and returns the item's masks, bounding boxes, and classes (He et al.; 2017). Figure 4 below shows the Mask R-CNN design for proposed system.

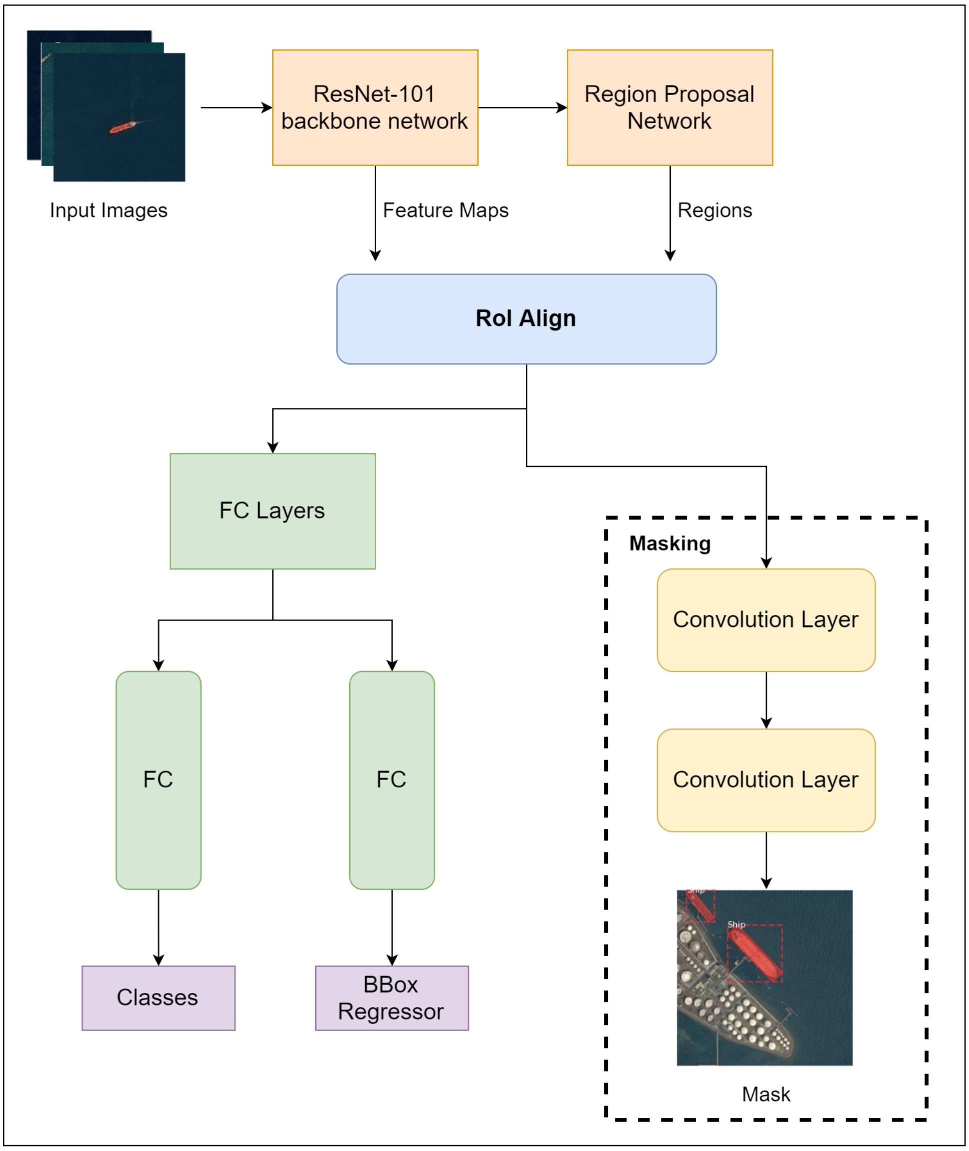


Figure 4: Mask R-CNN with ResNet-101 backbone architecture

The Mask R-CNN model works as follows:

4.1.1 Backbone Network

The raw SAR picture is sent to ResNet-101, a backbone network that acts as a feature extractor. Initial layers recognize low level characteristics such as boundaries and edges, while successive layers identify top level elements in the image (in this case, ships). The picture is changed from 1024 × 1024 × 3 (RGB) to a 32 × 32 × 2048 feature vector as it passes across the backbone network. This feature vector is then used as the input for the subsequent stages. The Figure 5 below depicts a rudimentary diagram of how a backbone network operates.

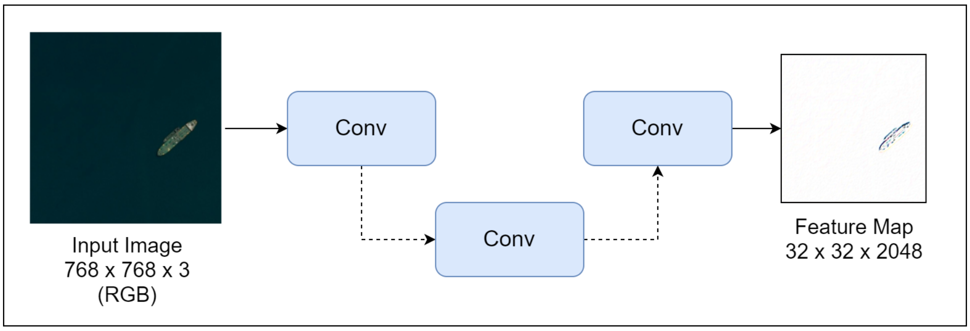


Figure 5: ResNet-101 backbone simple illustration

4.1.2 Region Proposal Network (RPN)

The RPN is a small neural net that reads a picture in a slide pane approach to discover areas with ships. The RPN reads feature map created by backbone network. The area that the RPN reads through is called as anchor. Anchors are the boxes spread across the image's surface. The RPN's convolutional structure handles the sliding frame, allowing it to read all regions in simultaneously (using GPU). It helps the RPN to save time by reusing the feature vectors and avoiding repeated computations. To every anchor, the RPN produces two outcomes: Anchor type (whether the anchor belongs to background or ship class) and bounding box refinement (a box generated around ship).

4.1.3 RoI Classifier and BBox Regressor

This phase is based on the RPN's recommended regions of interest (ROIs). It produces two outcomes for every ROI, same as the RPN: Class and BBox Refinement. These results are more precise than those provided by RPN.

4.1.4 RoI Align

Varying input size is difficult for classifiers to deal. A specific input size is usually needed. The ROI boxes, however, can vary in dimension because of the RPN's bounding box refining stage. This is when ROI Pooling enters the picture. Cropping and resizing a section of a feature vector to a definite size is known as ROI align.

4.1.5 Masking

We can generate a faster R-CNN network for object recognition if we end our framework at the RoI Align phase. The mask network is a new feature presented by the (He et al.; 2017). The mask section is a CNN that creates masks for positive regions identified by the ROI classifier.

5 Implementation

This chapter describes the research's entire execution. It covers the setup process, data processing stages, and model execution, as well as the tools utilized.

5.1 Environment Setup

System configuration: Windows 10 OS, 8GB RAM, NVIDIA GeForce GTX 1650Ti graphics.
Programming language: Python 3.6
IDE: Google Colab

5.2 Data Processing

The data is downloaded from Kaggle's 'Airbus Ship Detection Challenge' which contains 2 folders: train_v2 and test_v2. Image annotations file is also downloaded which contains ImageId and its respective encoded pixels information in RLE format. This data is uploaded on Google drive which is then accessed using google.colab's drive library[2] and files are handled using shutil library of python[3]. The train_v2 folder was used for model training and testing whereas test_v2 folder was used for final validations. Before splitting data for training and testing, all corrputed images and images with no ships were filtered out. The data was divided in 80:20 ratio for model training and testing. ImageAug package was used to perform data augmentation[4]. Augmentations such as random rotations, horizontal and vertical flipping, contrast and brightness manipulation, image blur and sharpen were performed on the dataset during model training.

5.3 Model Training

The Mask R-CNN model was trained using COCO weights[5]. We used SGD optimizer during training because as per (He et al.; 2017) SGD optimizer is ideal for training Mask R-CNN model. We trained model heads for 2 epochs without applying any augmentation, then we trained all layers of model with augmentation for 12, 8, 14 and 14 epochs respectively using different learning rates as mentioned in below Table 1:

Epoch	**Learning Rate**
1-14	1e-3
15-22	1e-4
23-36	5e-5
37-50	1e-5

Table 1: Learning rate decay

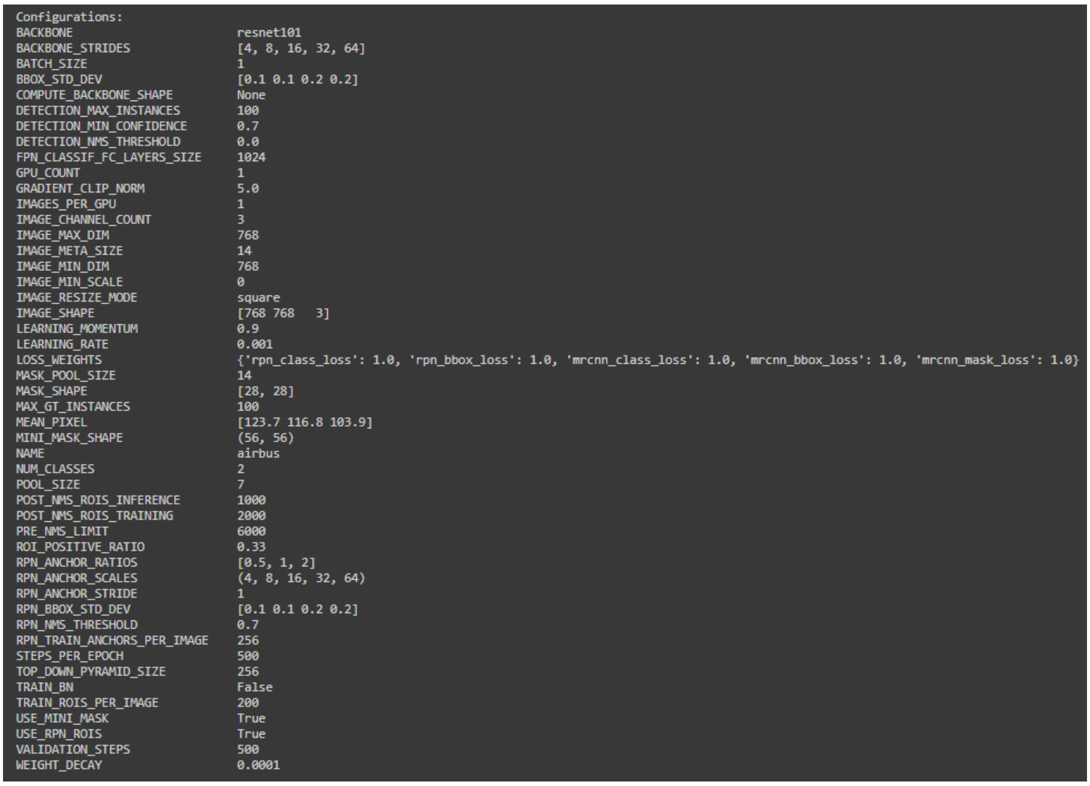
The learning rates were decided after a lot of trail runs and examination of the losses produced by the model. After training the model for 50 epochs, the best epoch was chosen on the basis of val_loss. The epoch with minimum val_loss was used for testing. It was made sure that model is not overfitting. The model was trained using different combinations of hyperparameter values in order to obtain good results. Figure 6 below shows the final configuration that was used to train our model.

[2] https://colab.research.google.com/notebooks/io.ipynb

[3] https://docs.python.org/2/library/shutil.html

[4] https://pypi.org/project/ImageAug/

[5] https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5



```
Configurations:
BACKBONE                        resnet101
BACKBONE_STRIDES                [4, 8, 16, 32, 64]
BATCH_SIZE                      1
BBOX_STD_DEV                    [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE          None
DETECTION_MAX_INSTANCES         100
DETECTION_MIN_CONFIDENCE        0.7
DETECTION_NMS_THRESHOLD         0.0
FPN_CLASSIF_FC_LAYERS_SIZE      1024
GPU_COUNT                       1
GRADIENT_CLIP_NORM              5.0
IMAGES_PER_GPU                  1
IMAGE_CHANNEL_COUNT             3
IMAGE_MAX_DIM                   768
IMAGE_META_SIZE                 14
IMAGE_MIN_DIM                   768
IMAGE_MIN_SCALE                 0
IMAGE_RESIZE_MODE               square
IMAGE_SHAPE                     [768 768   3]
LEARNING_MOMENTUM               0.9
LEARNING_RATE                   0.001
LOSS_WEIGHTS                    {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                  14
MASK_SHAPE                      [28, 28]
MAX_GT_INSTANCES                100
MEAN_PIXEL                      [123.7 116.8 103.9]
MINI_MASK_SHAPE                 (56, 56)
NAME                            airbus
NUM_CLASSES                     2
POOL_SIZE                       7
POST_NMS_ROIS_INFERENCE         1000
POST_NMS_ROIS_TRAINING          2000
PRE_NMS_LIMIT                   6000
ROI_POSITIVE_RATIO              0.33
RPN_ANCHOR_RATIOS               [0.5, 1, 2]
RPN_ANCHOR_SCALES               (4, 8, 16, 32, 64)
RPN_ANCHOR_STRIDE               1
RPN_BBOX_STD_DEV                [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD               0.7
RPN_TRAIN_ANCHORS_PER_IMAGE     256
STEPS_PER_EPOCH                 500
TOP_DOWN_PYRAMID_SIZE           256
TRAIN_BN                        False
TRAIN_ROIS_PER_IMAGE            200
USE_MINI_MASK                   True
USE_RPN_ROIS                    True
VALIDATION_STEPS                500
WEIGHT_DECAY                    0.0001
```

Figure 6: Mask R-CNN training configuration

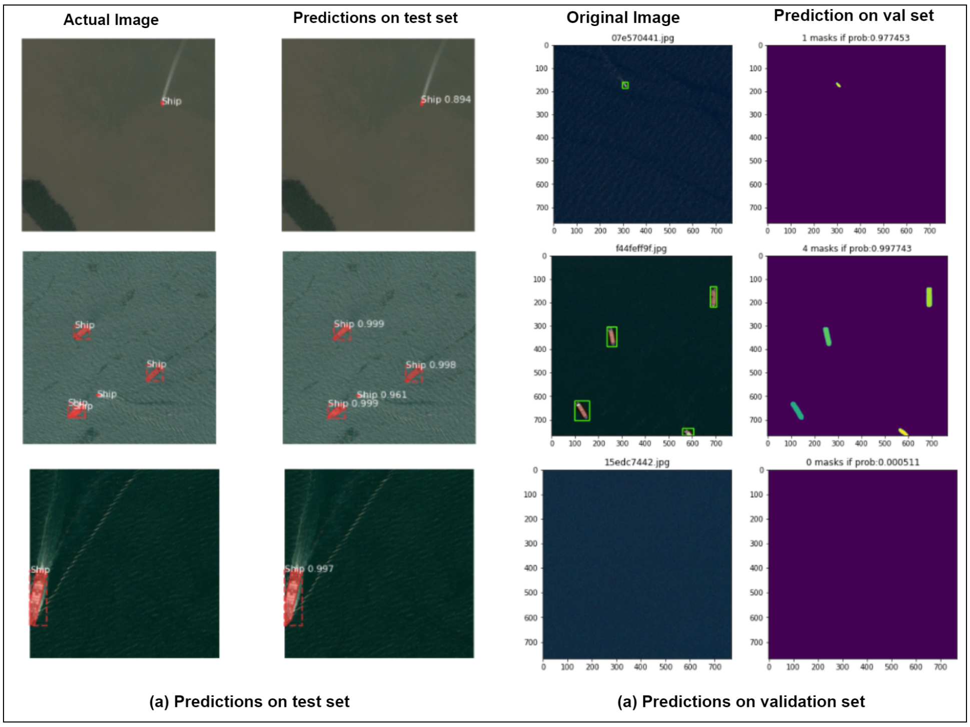
6 Evaluation and Discussion

6.1 Model Evaluation

In this section, we will discuss different evaluations that were carried out after successful execution of our model.

6.1.1 Results

The model was successfully able to detect ships during testing and validation phase. Below are a few sample results that were produced by our model during testing and validation. It can be seen from Figure 7, our model is able to small as well as large ships in test dataset. It is also able to detect multiple ships in one image. Ship detection on validation set is also showing satisfactory results. For each successful detection, appropriate masks are being produced. The model is producing a few false alarms for water streams and rectangular surfaces in few images.



(a) Predictions on test set

(a) Predictions on validation set

Figure 7: Predictions on testing dataset

6.1.2 Losses

Whenever an epoch runs, the model maintains a list of different losses experienced during training. Below are the different losses returned by each epochs:

- **loss:** It is total training loss experienced by model during an epoch.
- **rpn_class_loss/val_rpn_class_loss:** It represents loss by RPN anchor classifier.
- **rpn_bbox_loss/val_rpn_bbox_loss:** It represents loss by RPN bounding box generator.
- **mrcnn_class_loss/val_mrcnn_class_loss:** It shows loss for classifier head of Mask R-CNN module.
- **mrcnn_bbox_loss/val_mrcnn_bbox_loss:** It shows loss by bounding box refinement component.
- **mrcnn_mask_loss/val_mrcnn_mask_loss:** It shows loss while creating masks in masking phase.
- **val_loss:** It is total validation loss experienced by model during an epoch

Figure 8 below shows different loss graphs for our model which is trained and tested without data augmentation. It can be seen that there is a variation in training loss and testing loss. We can see that Train_loss is decreasing smoothly after each epoch however, Val_loss is remaining constant between 1.00 to 1.20. This also indicates that model might start to overfit if we continue to train it for more epochs.

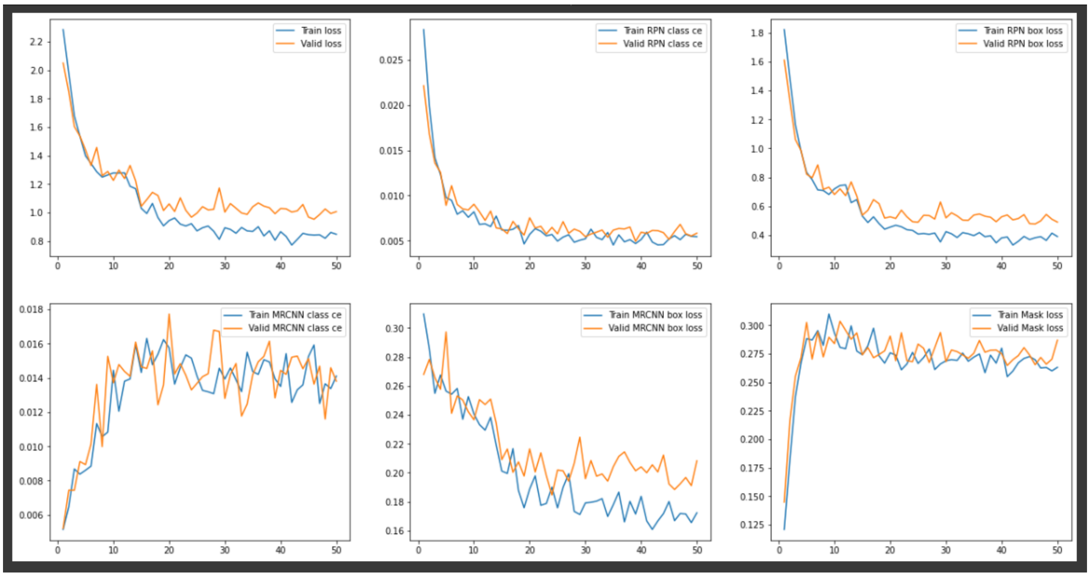


Figure 8: Mask R-CNN Loss Graph (without using data augmentation)

On the other hand, Figure 9 below shows different loss graphs for our model while using data augmentation. It can be clearly seen that overall model loss is much less when we train and test our model using data augmentation.

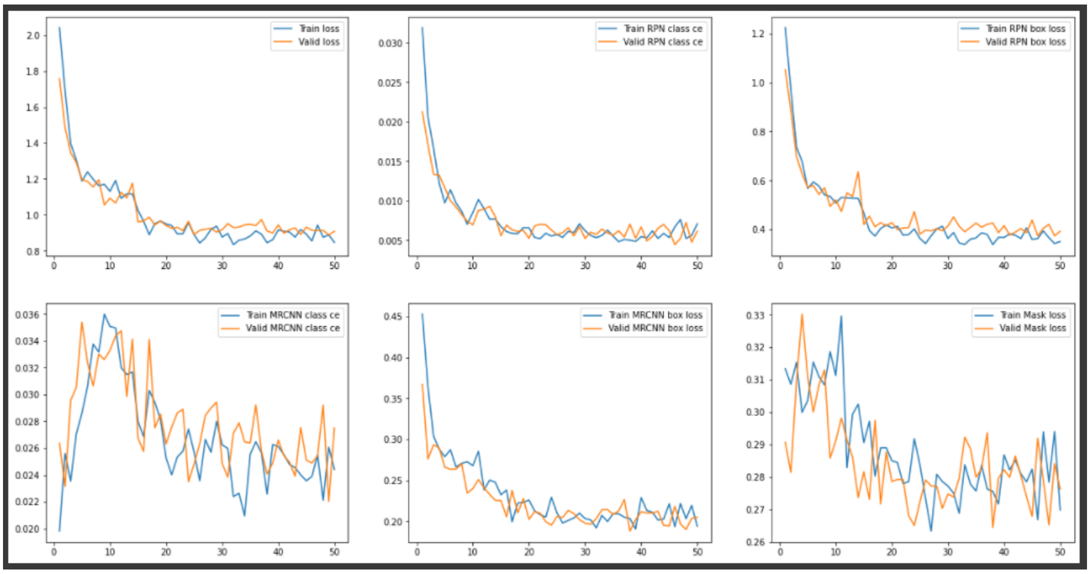


Figure 9: Mask R-CNN Loss Graph (using data augmentation)

6.1.3 Mean Average Precision (mAP)

Mean Avaerage Precision is a very popular metric that is used by almost all researchers to evaluate the performance of their object detection model. The mAP is calculated using below formula[6]:

[6] https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

Here,
Q : Total number of queries in given set.
AveP(q) : Average precision (AP) for given query q in the set.

For each query, q, we compute its associated AP, and afterwards take the average of all these AP values to get a single number called mAP. If mAP score is between 0.5 to 1.0, it denotes that our model is balanced, which means it can detect objects correctly with a few false alarms. Whereas, mAP score which is below 0.5 denotes model is unbalanced and needs tuning.

Our Mask R-CNN model yielded mAP score as 0.489 when no augmentation was used and 0.7152 when we applied data augmentation which shows that our model is well balanced.

6.2 Discussion

The research aims to evaluate performance of proposed Mask R-CNN model with ResNet-101 as a backbone network with data augmentation against modified version of Mask R-CNN implemented by (Nie et al.; 2020). The major difference between proposed model and the model implemented in (Nie et al.; 2020) is, we designed our network with ResNet-101 as backbone and also used different data augmentations to expand the size of dataset.

We compare the models on the basis of mAP as it is the best evaluation metrics used for object detection tasks. The model in (Nie et al.; 2020) was executed for 350,000 iterations with no data augmentations performed. They also tuned a few hyperparameters such as learning rate, RPN anchor scales, weight decay and aspect ratio which allowed them to achieve mAP of 76.10%. In this research, we ran our model for 25,000 iterations which allowed it to achieve mAP of 48.9% in the first run. However, after fine tuning our model using different methods such as applying data augmentation, reducing learning rate after a few epochs (learning rate decay), changing RPN anchor scales, changing batch size in configuration and changing the number of epochs, mAP of 71.52% was achieved. This mAP is lower than that achieved in (Nie et al.; 2020) but we were able to prove that data augmentation can help a deep learning model to improve its performance. Another challenge that we faced while implementing proposed model is resource limitations, Google colaboratory provides a limited RAM, disk and GPU space to run the python codes, after running our model for 25,000 iterations, the resources were getting exhausted or session was getting terminated which made us unable to continue further. The results that we were able to achieve are satisfactory and the implemented model can detect ships precisely with minimum false alarms. A few changes in model design can be made such as we could continue to tune hyperparameters, try using different optimizers for training and modifying inner layers of Mask R-CNN. These changes might help us to improve the current performance of our model.

7 Conclusion

Detecting maritime targets using Mask R-CNN and data augmentation was main objective of this research. Our model was developed using Mask R-CNN with ResNet-101 as a backbone network to detect ships from SAR images. We also used different data augmentation techniques such as rotation, flipping, contrast adjustment, brightness enhancement to train our model. We evaluated the outcomes achieved by training the model with and without using data augmentation. The results showed that using data augmentation can help our model to improve its performance. We were able to achieve mAP of 71.52% after some hyperparameter tuning and data augmentations. Due to resource limitations, the proposed model was not able to perform better than Mask R-CNN + FPN varient (Nie et al.; 2020). But there is not much of a difference between the mAP values of the implemented model and the model proposed in (Nie et al.; 2020).

The future work involves running the model with more resources and modifying the design of proposed model in order to improve its performance. Different optimizers can be used to reduce the model loss during training. The model can also be extended to detect ground targets along with maritime targets which could be a game changer in SAR-ATR.

8 Acknowledgment

I would like to express my gratitude to Prof. Hicham Rifai for his constant support, suggestions, and reviews. I am grateful to him for always encouraging and guiding me to improve my research. I would like to express my gratitude to my family and friends for their constant support.

References

Chen, H. and Cao, P. (2019). Deep learning based data augmentation and classification for limited medical data learning, *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pp. 300–303.

Ding, B., Wen, G., Huang, X., Ma, C. and Yang, X. (2017). Data augmentation by multilevel reconstruction using attributed scattering center for sar target recognition, *IEEE Geoscience and Remote Sensing Letters* **14**(6): 979–983.

Dong, M., Cui, Y., Jing, X., Liu, X. and Li, J. (2019). End-to-end target detection and classification with data augmentation in sar images, *2019 IEEE International Conference on Computational Electromagnetics (ICCEM)*, pp. 1–3.

Fu, Z., Zhang, F., Yin, Q., Li, R., Hu, W. and Li, W. (2018). Small sample learning optimization for resnet based sar target recognition, *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2330–2333.

Gui, Y., Li, X., Xue, L. and Lv, J. (2019). A scale transfer convolution network for small ship detection in sar images, *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 1845–1849.

Guohang, L., Shibin, Z., Haozhe, T., Lu, Y., Lu, J. and Yuanyuan, H. (2020). Easy data augmentation method for classification tasks, *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 166–169.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017). Mask r-cnn, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988.

Hou, Z., Cui, Z., Cao, Z. and Liu, N. (2020). An integrated method of ship detection and recognition in sar images based on deep learning, *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1225–1228.

Kechagias-Stamatis, O. and Aouf, N. (2019). Fusing deep learning and sparse coding for sar atr, *IEEE Transactions on Aerospace and Electronic Systems* **55**(2): 785–797.

Kun, J. and Yan, C. (2020). Sar image ship detection based on deep learning, *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)*, pp. 55–59.

Li, Y., Ding, Z., Zhang, C., Wang, Y. and Chen, J. (2019). Sar ship detection based on resnet and transfer learning, *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1188–1191.

Nie, X., Duan, M., Ding, H., Hu, B. and Wong, E. K. (2020). Attention mask r-cnn for ship detection and segmentation from remote sensing images, *IEEE Access* **8**: 9325–9334.

Vorhies, W. (2016). Crisp-dm: a standard methodology to ensure a good outcome.
URL: *https://www.datasciencecentral.com/pro
les/blogs/crisp-dm-a-standard- methodology-to-ensure-a-good-outcome*

Wu, T. D., Yen, Y., Wang, J. H., Huang, R. J., Lee, H. W. and Wang, H. F. (2020). Automatic target recognition in sar images based on a combination of cnn and svm, *2020 International Workshop on Electromagnetics: Applications and Student Innovation Competition (iWEM)*, pp. 1–2.

Xue, R. and Bai, X. (2019). 2d-temporal convolution for target recognition of sar sequence image, *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, pp. 1–4.

Yang, T., Zhu, J. and Liu, J. (2019). Sar image target detection and recognition based on deep network, *2019 SAR in Big Data Era (BIGSARDATA)*, pp. 1–4.

Zheng, C., Jiang, X. and Liu, X. (2019). Semi-supervised sar atr via multi-discriminator generative adversarial network, *IEEE Sensors Journal* **19**(17): 7525–7533.

Zhu, H., Hon, M., Wong, W., Leung, R., Lin, N. and Lin, K. (2020). Ground target classification from sar image with the pixel complement for target shape, *2020 IEEE SENSORS*, pp. 1–4.

Zhu, H., Lin, N., Leung, H., Leung, R. and Theodoidis, S. (2020). Target classification from sar imagery based on the pixel grayscale decline by graph convolutional neural network, *IEEE Sensors Letters* **4**(6): 1–4.