VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgavi-590 018, Karnataka, India

A DISSERTATION REPORT

On

DETECTION OF SHIPS IN SATELLITE IMAGERY USING DEEP LEARNING TECHNIQUES

Submitted in Partial Fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

MOHITH AN	**1MV19CS072**
REHAN NADEEMULLA	**1MV19CS092**
RAHUL GHOSH	**1MV19CS090**
MD SHAHID	**1MV19CS068**

Carried out at
Department of CSE,
Sir M. Visvesvaraya Institute of Technology

Under the guidance of
Dr. SREENIVASA B C
Associate Professor
Dept. Of CSE
Sir MVIT

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BENGALURU-562157
2022-2023

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
Bengaluru - 562157
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled **"**Detection of Ships In Satellite Imagery Using Deep Learning Techniques**"** is a bonafied work carried out by **Mohith AN (1MV19CS072), Rehan Nadeemulla (1MV19CS092), Rahul Ghosh (1MV19CS090), Md Shahid (1MV19CS068)** in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during the year **2022-2023.** It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements with respect to project work prescribed for the Bachelor of Engineering degree.

.....	.....	.....
Signature of Guide	Signature of HOD	Signature of Principal
Dr. Sreenivasa B C	**Dr. Anitha.T.N**	**Prof. Rakesh S G**
Associate Professor	Professor & HOD,	Principal SMVIT,
Dept. of CSE, SMVIT	Dept. of CSE, SMVIT	Bengaluru

External Examiners:

Name of the Examiners	**Signature with Date**
1.	
2.	

DECLARATION

We **Mohith AN, Rehan Nadeemulla, Rahul Ghosh, Md Shahid** Student of VIII semester B.E in Computer Science and Engineering at Sir M. Visvesvaraya Institute of Technology, Bengaluru, hereby declare that this dissertation work entitled **"**DETETCION OF SHIPS IN SATELLITE IMAGERY USING DEEP LEARNING TECHNIQUES**"** has been carried out at Dept. of CSE, Sir M. Visvesvaraya Institute of Technology under the guidance of guide **Dr. Sreenivasa B C, Associate Professor, Dept. of CSE, Sir M. Visvesvaraya Institute of Technology, Bengaluru** and submitted in the partial fulfillment for the award of degree Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2022-2022. We further declare that the report had not been submitted to any other university forthe award of any other degree.

Place: Bengaluru

Date:

Mohith AN (1MV19CS072)

Rehan Nadeemulla (1MV19CS092)

Rahul Ghosh (1MV19CS090)

Md Shahid (1MV19CS068)

ABSTRACT

In the commercial/military realm, ship detection has a variety of applications, and offshore security is an increasingly important need for countries around the world. For many private and public institutions, it is essential to comprehend marine activities on water bodies and at sea. Illegal, Unregulated, and unreported fishing and human trafficking must be prevented or deterred by the monitoring and regulation of activities like fishing, cargo transportation, passenger travel, and recreational traffic and it aids in preventing illegal activity. The numerous disruptions and sounds in these kinds of photos make ship detection one of the most difficult tasks. Because ships come in a variety of shapes and sizes, it might be challenging to identify a pattern or any regularity in these pictures. When there are merely ships of different types in the ocean, it is comparatively simpler in a homogenous environment. The issue, however, becomes tenfold in heterogeneous environments, which include additional components like beaches, harbors, vessels, rocks, islands, etc. The advantage of Synthetic Aperture Radar (SAR) in remote sensing is that it offers continuous coverage during all weather conditions.

ACKNOWLEDGEMENT

It gives us immense pleasure to express our sincere gratitude to the management of **Sir M. Visvesvaraya Institute of Technology,** Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

We would also like to convey our regards to **Prof. Rakesh S G,** Principal, Sir MVIT for providing us with the infrastructure and facilities needed to develop our project.

Heartfelt and sincere thanks to **Dr. T. N. Anitha,** HOD, Dept. of CSE, for her suggestions, constant support and encouragement.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide **Dr. Sreenivasa B C,** Associate Professor, Dept. of CSE, for his invaluable help and guidance.

We would also like to thank the staff of Department of Computer Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would liketo thank all our friends for their help and suggestions without which completing this project would not have been possible.

Mohith AN (1MV19CS072)
Rehan (1MV19CS092)
Rahul Ghosh (1MV19CS090)
Md Shahid(1MV19CS068)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER - 1

INTRODUCTION

1.1 Overview

In remote sensing image classification, machine learning has proven to be an essential research field. Data for picture classification are composed of numerous sample s that are each represented by numerous datasets. High levels of accuracy in classification and training performance are therefore extremely difficult to achieve. Machine learning approaches have been widely utilized to create precise classification models. To boost the detection accuracy, a number of strategies have been put forth in this area. Even other hybrid methods are put forth; however, they are incompatible with extreme weather since they could result in erroneous detection.

Ship Detection is the basic need in maritime security; It may be used to look for lost ships, commercial and also military ships. Though Automated Identification Systems exists in Ships, they can be manually disabled to avoid identification. Therefore, there is a need to develop and automated ship detection system which can perform its task in real time and is also controlled only by the authorized personnel.

The advancements in deep learning and publicly available remote sensing data when combined, give an effective solution to this problem. Different types of images, such as optical images, SAR images, and aerial photography images, are commonly employed to solve this problem. Though SAR images are the mostly commonly used, optical images can also be efficient.

Despite not being perfect in all circumstances, these cutting- edge techniques were reliable to a point. This is the rationale behind considering deep learning. In order to achieve high levels of accuracy, CNN models can be trained using thousands of instances, including all possible worst-case scenarios. The main objective is to detect objects, and in this case, a ship is the object. The CNN neural network performs excellently in this regard. A deep learning model represents data at different levels of abstraction by incorporating multiple layers of processing. Through the combination of convolutional neural net- works (CNN) and powerful graphical

processing units (GPUs), it has been able to achieve astounding success in the object classification and detection. Image classification is crucial (Wen et al., 2015) in satellite imagery interpretation. Moreover, in numerous fields, including land management, urban planning, environmental research and monitoring, and the early detection of natural disasters. There are a number of applications for satellite images of ships with high resolution, including maritime monitoring, illegal fishing, secure and efficient transport in harbours, military purposes, and many more.

In this project, a model has been developed using ResNet50 architecture pre-trained on ImageNet dataset. A comparative study has been performed to note the behavior of the model with and without transfer learning.

1.2 Problem Statement

"This is a proposal to train and analyze satellite images dataset to detect ships using Deep Learning Techniques"

Input: First, a dataset titled "Ships in satellite imagery" from Kaggle. This dataset consists of 4000 optical aerial images. Second, MASATI V2 dataset is used.

Output: Accurate result which says whether an image contains ship or not.

1.3 Significance and Relevance of Work

Vessel Detection is a very important part in maritime security application and remote sensing data is a central source of data for this task. Optical satellite images and Synthetic Aperture Radar (SAR) images can be used to solve this problem. Both systems have their own advantages and limitations. SAR sensors cover very large areas and they are weather independent, hence, very effective for ship detection tasks. However, they are limited in terms of ship classification/identification without help of auxiliary datasets. Optical satellite images can provide more information on the content and also details about the object (vessel) itself, such as texture, shape and colour. 0.3m per pixel. Such level of detail is sufficient to distinguish different kind of ship features like bridges, cranes, etc. Applying deep learning techniques on this remote sensing data can help identify the various objects on the sea surface. Though many

ship detection methods have been proposed before, this task still poses a great challenge due to the existence of uncertainties such as light, disruptors, the density of the ship and so on.

Ship detection using satellite imagery holds significant significance and relevance across various domains and industries. Here are some key points highlighting its importance:

- **Maritime Security**: Ship detection helps in enhancing maritime security by providing real-time information about ship movements. It enables monitoring of vessel traffic and identification of potentially suspicious or unauthorized activities, such as illegal fishing, smuggling, piracy, or human trafficking. Timely detection allows for appropriate response and intervention to maintain maritime safety and security.
- **Search and Rescue Operations**: Satellite-based ship detection plays a crucial role in search and rescue operations during emergencies at sea. By analysing satellite imagery, authorities can identify distressed vessels, locate survivors, and coordinate rescue efforts efficiently. It improves response times, saves lives, and enhances overall maritime safety.
- **Environmental Monitoring**: Ship detection aids in environmental monitoring and protection of marine ecosystems. Satellites can identify and track vessels involved in illegal dumping, oil spills, or other hazardous activities that may harm the environment. Early detection enables timely intervention, minimizing the ecological impact and facilitating prompt clean-up operations.
- **Maritime Traffic Management**: Satellite-based ship detection assists in efficient maritime traffic management. By monitoring ship movements and analysing traffic patterns, authorities can optimize port operations, prevent congestion, and ensure smooth navigation in busy shipping lanes. This improves overall maritime transportation efficiency and reduces the risk of accidents.
- **Economic and Trade Analysis**: Ship detection provides valuable data for economic and trade analysis. By tracking vessel movements and monitoring port activities, satellite imagery helps in assessing trade flows, analysing import-export patterns, and evaluating the economic impact of shipping activities. This information is crucial for governments, businesses, and financial institutions involved in international trade.
- **Border Surveillance**: Ship detection using satellite imagery contributes to border surveillance and coastal defence. It allows monitoring of territorial waters, identification of potential threats, and prevention of unauthorized border crossings. By

integrating satellite data with other surveillance systems, authorities can enhance their situational awareness and security capabilities.

- **Insurance and Risk Assessment**: Ship detection aids insurance companies and risk assessors in evaluating maritime risks. Satellite imagery helps assess vessel conditions, monitor shipping routes, and identify areas prone to accidents or natural disasters. This information assists in determining insurance premiums, underwriting policies, and risk management strategies.
- **Illegal Fishing Detection**: Satellite-based ship detection is valuable in combating illegal fishing activities. By monitoring fishing vessels and their activities, authorities can identify unauthorized fishing zones, enforce fishing regulations, and protect marine resources. This helps promote sustainable fishing practices and conserve marine biodiversity.

1.4 Objectives

The objectives of ship detection using Convolutional Neural Networks (CNN) and ResNet (a specific type of CNN architecture) can be summarized as follows:

- **Ship Classification**: The primary objective is to develop a model that can accurately classify whether an object in a satellite image is a ship or not. CNNs, with their ability to automatically learn relevant features from images, are well-suited for this task. ResNet, in particular, with its deep architecture and skip connections, helps overcome the challenge of vanishing gradients in deep networks and enables effective feature extraction.

- **Ship Localization**: In addition to classifying ships, another objective is to localize the ship's position in the satellite image. This involves identifying the bounding box or pixel-level segmentation of the ship within the image. By incorporating localization techniques into the CNN or ResNet model, it becomes possible to precisely determine the ship's location, facilitating further analysis and decision-making.

- **Ship Counting**: Ship detection also involves estimating the number of ships present in a satellite image or a given region. By training a CNN or ResNet model on labeled data that includes ship counts, the objective is to develop an algorithm that can accurately

count the ships present in new, unseen images. This can be valuable for maritime traffic management, surveillance, and other applications.

- **Accuracy and Robustness**: The objective is to achieve high accuracy and robustness in ship detection. This involves training the CNN or ResNet model on a diverse and representative dataset, which includes various ship types, orientations, sizes, and environmental conditions. The model should be capable of generalizing well to different scenarios and maintaining its performance under varying lighting conditions, weather conditions, and image quality.

- **Real-time or Near-real-time Detection**: Another objective is to develop ship detection models that can provide real-time or near-real-time results. This is particularly important for time-sensitive applications such as maritime security, where timely detection and response are critical. CNN and ResNet models should be designed and optimized to provide efficient and fast ship detection without compromising accuracy.

- **Generalization to Different Satellite Sensors**: Ship detection models using CNN and ResNet should be designed to generalize well across different satellite sensors and resolutions. This ensures that the trained models can be applied to various satellite imagery sources and be adapted to different spatial resolutions, spectral bands, and imaging characteristics.

1.5 Methodology

ResNet50

Residual network, most commonly known as ResNet, is a special type of neural network. It stacks architectural structures called Residual blocks, one over the other to form a network. A technique called skip connections, connects activation layers to further layers by skipping some layers in between to form a residual block. This network has evolved to solve the vanishing gradient problem.

ResNet50 is a pre-trained 50-layer deep learning model trained on ImageNet, to aid in image classification. It can classify objects into 1000 categories.

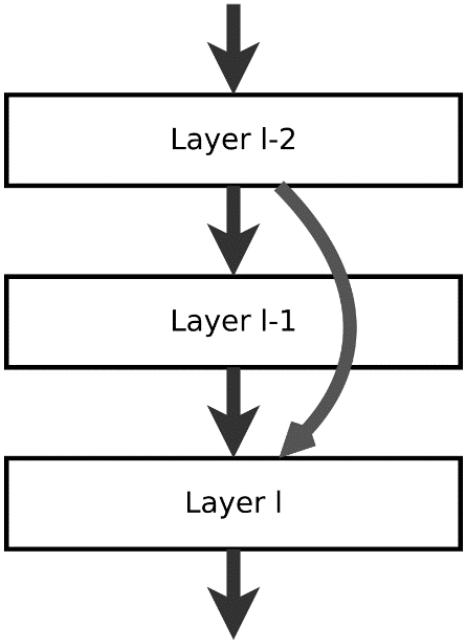


Fig 1.1: Canonical form of a residual neural network. A layer $\ell - 1$ is skipped over activation from $\ell - 2$.

Transfer Learning

Transfer learning is a deep learning application whereby a fully developed network is re-trained with a pre-trained network which serves as a starting point to learn a new task for enhanced convergence and performance. The features of the pre-trained network are transferred via transfer learning to a new task using a smaller number of training images, features and parameters. These features are applied to a wide range of other similar tasks to reduce the training time with improved accuracy. In this research work, we introduce transfer learning using ResNet to pretrain the network.

Using the MASATI V2 dataset for ship detection using CNN and ResNet, the methodology can be outlined as follows:

1. **Data Collection**: Obtain the MASATI V2 dataset, which consists of satellite images labelled with ship and non-ship annotations. The dataset specifically focuses on ship detection and includes a variety of ship types and sizes.
2. **Data Pre-processing**: Pre-process the satellite images from the MASATI V2 dataset to make them suitable for training the CNN and ResNet models. This may involve resizing the images to a consistent resolution, normalizing pixel values, and augmenting

the dataset with techniques such as rotation, scaling, or flipping to increase its variability.

3. **Training-Validation Split**: Split the MASATI V2 dataset into training and validation sets. The training set is used for training the CNN and ResNet models, while the validation set is used for monitoring performance and tuning hyperparameters during training.
4. **Model Architecture**: Design the CNN and ResNet architectures for ship detection. Consider the complexity of the ship detection task and available computational resources when selecting the specific architecture. CNNs typically consist of multiple convolutional layers for feature extraction, followed by pooling layers and fully connected layers for classification. ResNet, with its skip connections, can be used to alleviate the vanishing gradient problem in deep networks.
5. **Transfer Learning**: Utilize transfer learning by leveraging pre-trained CNN and ResNet models trained on large-scale image datasets like ImageNet. Use the pre-trained models as a starting point and fine-tune them on the MASATI V2 dataset to learn ship-specific features.
6. **Model Training**: Train the CNN and ResNet models using the labelled training data from the MASATI V2 dataset. During training, the models learn to extract relevant features from the satellite images and make accurate ship/non-ship predictions. This is accomplished by minimizing a suitable loss function, such as binary cross-entropy, using backpropagation and gradient descent.
7. **Hyperparameter Tuning**: Fine-tune the hyperparameters of the CNN and ResNet models to optimize their performance on the MASATI V2 dataset. Adjust hyperparameters such as learning rate, batch size, number of layers, filter sizes, activation functions, and regularization techniques based on the validation set's performance.
8. **Model Evaluation**: Evaluate the trained CNN and ResNet models using the validation set from the MASATI V2 dataset to assess their performance. Use evaluation metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC) to measure the models' effectiveness.
9. **Testing**: Apply the trained CNN and ResNet models on unseen satellite images from the MASATI V2 dataset for ship detection. Pass the test images through the models to obtain ship/non-ship predictions. Employ post-processing techniques like thresholding or non-maximum suppression to refine the detection results if necessary.

1.6 Organization of the Report

1. **Introduction:**

Background, problem statement, and objectives.

2. **Literature Review:**

Review of relevant studies on green infrastructure and prediction models.

3. **Data Collection and Preprocessing:**

Description of data sources and preprocessing steps.

4. **Methodology:**

Explanation of the selected prediction model and steps involved.

5. **Model Development and Evaluation:**

Presentation and evaluation of the developed model.

6. **Results and Analysis:**

Presentation and analysis of the predicted suitability maps.

7. **Implementation and Recommendations:**

Discussion of practical implications and recommendations for stakeholders.

8. **Limitations and Future Work:**

Identification of limitations and suggestions for future improvements.

9. **Conclusion:**

Summary of findings and project contributions.

10. **References:**

List of cited sources

CHAPTER 2
LITERATURE SURVEY

CHAPTER – 2

LITERATURE SURVEY

2.1 Multiclass vessel detection from high resolution optical satellite images Based on deep neural networks.

Publication: IEEE, IGARSS 2019.

Algorithm/ Techniques: Deep Convolutional Neural networks (DCNN) and principal component analysis (PCA).

Findings: CNN model "MobileNet" is used to classify images on the basis whether they contain ship/ship parts or not. Next, resulting in image subsets are processed with complex object detection model "Faster R-CNN ResNet-101" to extract locations, classes, and north oriented minimum-maximum bounding boxes of the ships. And finally, with the help of PCA, vessel heading is estimated.

Limitations/ Scope of Future Work: Further research will involve tests with different network architectures and settings. The training dataset is constantly enlarging with perspective to increase the number of object classes. Additional subject of research is the inclusion of Automatic Identification System (AIS) data into the training as well as in the detection processes.

2.2 Ship Detection from Optical Satellite Images with Deep Learning.

Publication: IEEE, RAST 2019.

Algorithm/ Techniques: Deep Learning Algorithm, TensorFlow Object Detection.

Findings: TensorFlow Object Detection Application Programming Interface (API) is trained using optical satellite images with ships to be used as an object detection API.

Limitations/ Scope of Future Work: The algorithm has been trained on optical images. It can further be extended to SAR images also.

2.3 Multi-Scale Ship Detection from SAR and Optical Imagery Via a More Accurate YOLOv3.

Publication: IEEE 2021

Algorithm/ Techniques: YOLO V3.

Findings: This paper proposes a new improvement on the yolov3 framework for ship detection in marine surveillance, based on synthetic aperture radar (SAR) and optical imagery.

Limitations/ Scope of Future Work: The dataset used here is a mixture of SAR images and Airborne images. The SAR images are taken from 102 Chinese Gaofen-3 images and 108 Sentinel-1 images. This makes the dataset of SAR images local to the region. A broader dataset can be taken, and model can be evaluated.

2.4 High-Speed Lightweight Ship Detection Algorithm Based on YOLO-V4 for Three-Channels RGB SAR Image.

Publication: MDPI, 2021.

Algorithm/ Techniques: Yolov4 model

Findings: This paper proposes a multi-channel fusion SAR Image processing method that makes full use of image information and the network's ability to Extract features; it is also based on the YOLO-V4 Framework for modelling architecture and training models.

Limitations/ Scope of Future Work: Future work should aim to find more useful information about ships to enrich the redundant colour channels of SAR images, and further optimize the network structure, especially for the detection performance of small and dense ships.

2.5 Ship Detection and Classification on Optical Remote Sensing Images Using Deep Learning.

Publication: ITM Web Of Conferences,2017.

Algorithm/ Techniques: CNN Model

Findings: A CNN model is trained to detect actual ships from all images. Further, using the CNN model, all the actual ships were classified into different ships.

Limitations/ Scope of Future Work: The CNN model can be further enhanced and evaluated on a bigger dataset of images.

2.6 Automatic Ship Detection of Remote Sensing Images from Google Earth in Complex Scenes Based on Multi-Scale Rotation Dense Feature Pyramid Networks.

Publication: Article

Algorithm/ Techniques: R-DPFN Framework.

Findings: This paper proposes a multi-scale rotation region detection method that can handle different complex scenes, detect intensive objects and reduce redundant detection region. A framework called Rotation Dense Feature Pyramid Networks (RDFPN) is used.

Limitations/ Scope of Future Work: Despite good results, more false alarms resulted in a much lower Precision for R-DFPN than Faster-RCNN and FPN. We need to explore how to effectively reduce false alarms in the future.

2.7 A deep learning approach to ship detection using satellite imagery.

Publication: Article, IOP Conf. Series: Earth and Environmental Science, 2020**.**

Algorithm/ Techniques: CNN Model.

Findings: A CNN based classifier − has been developed based on DenseNet architecture. The algorithm has been optimized to achieve optimal results.

Limitations/ Scope of Future Work: This algorithm can be further fine-tuned by considering other types of convolutional neural network architecture to increase detection accuracy.

2.8 Machine Learning Based Ship Detection Technique

Publication: SERSC, 2020.

Algorithm/ Techniques: CNN Model.

Findings: This paper presents a machine learning based Ship Detection technique from high resolution satellite images with CNN.

Limitations/ Scope of Future Work: More algorithms can be explored with this technique to get better results

2.9 Ship Detection from Despeckled Satellite Images using Deep Learning

Publication: IEEE 2022

Algorithm/Techniques: CNN Model

Findings: Here a CNN based ship detection model has been proposed. The proposed model is evaluated using dataset including 4000 images with the resolution of 80X80 RGB satellite images labelled either “ship” or “not ship.” In order to train the model, the images are despeckled using Gaussian and Average filters.

Limitations/ Scope of Future Work - The accuracy can be further enhanced using other despeckling approaches in combination with different architecture of CNN model.

2.10 Deep Learning-Based Automatic Detection of Ships: An Experimental Study Using Satellite Images.

Publication: MDPI, 2022

Algorithm/ Techniques: Yolo Algorithm

Findings: Here YOLOv3 and YOLOv4 were tested and YOLOv5 was used in an experimental ship detection task. The algorithms' performance was compared to see which method gave the best results in ship detection. For training, testing, and validation, the Airbus Ship Challenge and Ships net satellite datasets were used, and the YOLO family algorithms were then evaluated on a PC. According to the findings, all three algorithms satisfy the criteria for the ship detection. Given the results, YOLOv5 has been chosen as the method with the highest accuracy in the detection of ships from satellite images

Limitations/ Scope of Future Work: To test the different sensors, such as SARs, under extreme conditions, where visible spectrum imagery is unavailable, e.g., at night, when it is cloudy, or if there is fog.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATIONS

CHAPTER – 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 System Requirements Specification

The System Requirements Specification document provides a comprehensive description of the requirements for the system being developed. It serves as a guide for the design and development process, outlining the functional and non-functional requirements that need to be met. The System Requirements Specification serves as a vital reference for the development team, ensuring that the system is designed and implemented to meet the specified requirements and achieve the desired outcomes.

3.2 Specific Requirement

3.2.1. Hardware Specification

- Processor: Intel(R) Core (TM) i3-70020U CPU@2.30GHz
- RAM: 4.00 GB DDR4
- Disk: 1 TB
- System Type: 64-bit Operating System

3.2.2. Software Specification

- OS: Windows 10 or above
- IDE: Google Colab
- Programming Language: Python 3.11
- Libraries: NumPy, pandas, keras, TensorFlow, OpenCV2, matplotlib, scikit-learn

3.3 Functional Requirements

- The deep learning ship classifier has been trained using a transfer learning approach with ResNet50 as pre-trained weights.
- The model must be able to classify a given image as having “ship” or “no-ship” using the pre-trained resnet50 model as feature extractor.
- The model must be able to minimise the False Positives as it may lead to unnecessary security drills.

- The False negatives must be minimised as it can pose a serious threat to the maritime security.

3.4 Non-Functional Requirements

- **PERFORMANCE:**
  - The classifier model must be able to classify an image with at least 75% in real-time system and the time taken for this must be minimum.
  - Optical satellite images may have a lot of noise due to weather in the form of clouds, haze etc., as all these can affect the clarity of the images acquired. The model must be able to classify images in these cases also.

- **EASE OF USE:**
  - The classifier model must be simple and easy to use so that it can be widely incorporated.

- **FLEXIBILITY:**
  - The optical images are taken in various circumstances due to which, the visibility, quality, sizes and resolutions. The classifier model must be able to classify these images efficiently.

- **SECURITY:**
  - Since the model is to be used in maritime security, it must be robust and highly secure. Any vulnerability can lead to serious and disastrous consequences.

- **COMPATIBLE:**
  - The model must be compatible to multiple platforms to facilitate its integration in real-time scenarios.

3.5 Performance Requirement

- The system should achieve a prediction accuracy of at least 80% for ship or no ship classification.
- The response time for generating predictions should be within 5 seconds for a single image.

- The system should be scalable to handle a large volume of data and perform predictions in real-time.

CHAPTER 4

SYSTEM ANALYSIS

CHAPTER – 4

SYSTEM ANALYSIS

4.1 Existing System

There are several existing deep learning-based systems for ship detection in satellite imagery.

- ShipNet: ShipNet is a deep learning-based ship detection system developed by the European Space Agency (ESA). It utilizes convolutional neural networks (CNNs) for ship detection and employs a two-stage architecture to refine ship locations. ShipNet has been trained on a large dataset of Sentinel-1 synthetic aperture radar (SAR) images.

- DeepShip: DeepShip is a ship detection system developed by the University of Science and Technology of China. It uses a combination of deep learning techniques, including deep convolutional neural networks (DCNNs) and region proposal networks (RPNs), to detect ships in satellite imagery. DeepShip has been trained on a diverse dataset of optical satellite images.

- Ship Detection and Tracking in Satellite Imagery (SDTSI): SDTSI is a ship detection system developed by researchers at the Indian Institute of Technology (IIT) Kharagpur. It combines deep learning models, such as Faster R-CNN and YOLO, with feature engineering techniques to detect and track ships in satellite imagery. SDTSI has been trained on a dataset comprising both SAR and optical satellite images.

- Ship Detection using Deep Convolutional Neural Networks (SDCNN): SDCNN is a ship detection system developed by researchers at the University of Wollongong, Australia. It employs a deep CNN architecture with multiple convolutional and pooling layers for ship detection in satellite imagery. SDCNN has been trained on a dataset of SAR images acquired by the RADARSAT-2 satellite.

- Ship Detection in Satellite Images (ShipDET): ShipDET is a ship detection system developed by researchers at the University of Trento, Italy. It utilizes a deep learning framework called FCRN-A, which combines a fully convolutional network (FCN) with a residual network (ResNet), for ship detection in satellite imagery. ShipDET has been trained on a large dataset of optical satellite images.

- Ship Detection and Tracking using Convolutional Neural Networks (SDTCNN): SDTCNN is a ship detection and tracking system developed by researchers at the University of Electronic Science and Technology of China. It utilizes a CNN-based architecture for ship detection in satellite images and employs a tracking algorithm to track ships across consecutive frames.

- Ship Detection in Satellite Imagery using Multi-Scale Convolutional Neural Networks (SD-MSCNN): SD-MSCNN is a ship detection system developed by researchers at the University of Oxford. It incorporates multi-scale convolutional neural networks to capture ships at different scales in satellite images. The system achieves high accuracy in ship detection and can handle large-scale datasets.

- Ship Detection in SAR Images using Fully Convolutional Neural Networks (SD-FCNN): SD-FCNN is a ship detection system developed by researchers at the University of Chinese Academy of Sciences. It utilizes fully convolutional neural networks to perform ship detection in synthetic aperture radar (SAR) images. The system leverages the capabilities of FCNNs to capture spatial information effectively.

- Ship Detection in Multispectral Satellite Imagery using Deep Learning (SDMS-DL): SDMS-DL is a ship detection system developed by researchers at the University of Notre Dame. It combines deep learning techniques with spectral analysis to detect ships in multispectral satellite imagery. The system takes advantage of the additional spectral bands to improve ship detection accuracy.

- Ship Detection in Optical Satellite Images using Attention Mechanism (SD-AM): SD-AM is a ship detection system developed by researchers at the University of Science and Technology Beijing. It incorporates an attention mechanism into the deep learning architecture for ship detection in optical satellite images. The attention mechanism helps the model focus on relevant ship-related features and improves detection performance.

4.1.1 Limitations

1. **Insufficient training data**: Deep learning models typically require large amounts of labelled data to achieve optimal performance. However, acquiring labeled ship data in

satellite imagery can be challenging and time-consuming. Limited training data may lead to suboptimal performance or difficulty in generalizing to new or unseen ship patterns.

2. **Imbalanced datasets**: Ship detection datasets often suffer from class imbalance, where the number of ship instances is significantly lower than non-ship instances. This class imbalance can impact the model's ability to accurately detect ships, as the model may become biased towards the majority class, resulting in high false negative rates.

3. **Variability in ship appearance**: Ships can vary significantly in terms of size, shape, orientation, and appearance due to factors such as viewpoint, lighting conditions, and occlusions. Deep learning models trained on a specific dataset may struggle to generalize to unseen ship variations, leading to decreased performance in real-world scenarios.

4. **False positives and false negatives**: Deep learning models for ship detection may produce false positives (detecting objects as ships when they are not) or false negatives (missing actual ship instances). False positives can arise due to misclassifying similar objects or confusing ship-like structures in the environment. False negatives can occur when ships are small, partially occluded, or have low contrast with the background.

5. **Limited interpretability**: Deep learning models are often considered black boxes, making it challenging to understand their decision-making process. Interpreting why a model detects or fails to detect a ship in a particular situation can be difficult. Lack of interpretability can hinder trust and understanding of the system's behavior, especially in critical applications such as maritime surveillance.

6. **Computationally expensive**: Deep learning models for ship detection often require substantial computational resources, including high-performance GPUs, to train and deploy. The inference time for real-time ship detection may be relatively high, limiting the system's efficiency for applications that require rapid response times.

4.2 Proposed System

The proposed system aims to enhance the existing ship detection system by leveraging advanced techniques and technologies. The system imports dataset images separately as ship

images and no-ship images. Data augmentation and data splitting into training and validation data is performed.

The proposed system consists of 3 models. First is the CNN model consists of 2D convolutional layer with 32 filters and a Max-pooling layer which reduces the spatial dimensions of the input, helping to extract important features while reducing the computational complexity. Then a Flatten layer is added to the model, which transforms the multidimensional feature maps into a one-dimensional vector. This is necessary to connect the convolutional layers to the fully connected layers. Two fully connected layers are added one with 16 neurons with ReLu activation function and another with 2 neurons with softmax function. Weights are freezed and model is compiled by specifying the loss function, optimizer, and evaluation metric.

Model 2 is the resent50 model imported from keras. In this model fully connected layers at the top of the network are not included and retrieves the output of the last layer of the ResNet50 model. This output represents the features learned by the model from the input images and the layer is not flattened. A new layer is added using the ResNet50 model's input and the modified output obtained from the Flatten layer. The resulting model represents the ResNet50 backbone with its original weights up to the last layer.

In model 3, resnet50 is fine-tuned by changing few parameters. First the trainable property is set to be true then 3 layers named ‘res5c_branch2b’, ‘res5c_branch2c’ and ‘activation_97’ is set to be trainable. Then this new layer will be added to the model

4.2.1 Advantages

1. **Deep Learning-Based Approach**: The system utilizes deep learning techniques, such as Convolutional Neural Networks (CNNs) and pre-trained models like ResNet, which have shown remarkable performance in image analysis tasks. Deep learning models can automatically learn and extract complex features from the satellite images, making them well-suited for ship detection.
2. **Large and Diverse Dataset**: The system combines two datasets, "Ships in satellite imagery" and MASATI-V2, providing a larger and more diverse training set. A large

dataset can help improve the model's ability to generalize and handle different ship variations, backgrounds, and environmental conditions.

3. **Data Augmentation**: The system employs data augmentation techniques to increase the number of training samples by applying various transformations to the existing images. Data augmentation can enhance the model's robustness, reduce overfitting, and improve generalization by exposing it to a wider range of image variations.
4. **Multiple Models**: The system explores multiple models, including a CNN model and variations of the ResNet model. This approach allows for comparison and selection of the most effective architecture for ship detection. Each model may have its strengths and weaknesses, and testing multiple models helps find the best-performing one.
5. **Transfer Learning**: The system leverages pre-trained models like ResNet, which are trained on large-scale image datasets (e.g., ImageNet). By utilizing pre-trained weights, the system can benefit from the learned features and knowledge of these models, potentially leading to faster convergence and improved performance.
6. **Fine-Tuning**: The system performs fine-tuning on the ResNet model, enabling it to adapt to the specific task of ship detection. Fine-tuning allows the model to adjust the pre-trained weights to better suit the characteristics of the ship detection problem, potentially enhancing its performance.
7. **Optimization Techniques**: The system utilizes the Adam optimizer, a popular optimization algorithm, which can improve the training process and convergence speed. Additionally, the use of techniques like non-trainable layers in the ResNet models can help stabilize training and prevent overfitting.
8. **Automatic Ship Detection**: Once trained, the system can automatically detect ships in satellite imagery. This can significantly reduce manual effort and provide timely and efficient ship detection capabilities for various applications, including maritime surveillance, security, and environmental monitoring.

CHAPTER 5

SYSTEM DESIGN

CHAPTER – 5

SYSTEM DESIGN

5.1 Project Modules

The ten modules that make up our suggested analysis were created in accordance with the necessary functional and non-functional requirements. The modules are individually described below with all the necessary details.

The **Data Collection module** is a crucial component of the ship detection machine learning project. It involves gathering relevant data that serves as input for the prediction model. The module encompasses various tasks, including data acquisition, cleaning, and preprocessing. Data sources may include satellite imagery from two datasets. The satellite imagery needs to be annotated with ground truth labels that indicate whether each image contains a ship or not. This can be a time-consuming task, and may involve using manual or semi-automatic annotation tools. The annotated data set needs to be split into training, validation, and testing sets. The training set is used to train the model, while the validation set is used to tune hyperparameters and prevent overfitting. The testing set is used to evaluate the model's performance on unseen data. This module plays a vital role in providing a solid foundation of reliable and comprehensive data for subsequent modules to generate accurate predictions for green infrastructure potential.

The **Feature Selection module** is a critical component of the ship detection machine learning project. Its purpose is to identify and select the most relevant features that significantly contribute to the prediction of ship. Normalize the pixel values of the input images to a common scale. This can help to improve the contrast of the images and make it easier for the model to distinguish between different objects. Resizing the input images to a common size. This can help to reduce the computational cost of the model and improve its performance. Data augmentation techniques such as random cropping, flipping, and rotation can be used to increase the diversity of the training data set and help the model learn robust features

The **Model Training module** is a crucial phase in the ship detection machine learning project. It involves training a prediction model using the selected features and historical data.

The hyperparameters of the CNN model, such as learning rate, batch size, and regularization, need to be tuned to optimize the performance of the model. This can be done using a validation set to prevent overfitting. A suitable loss function, such as binary cross-entropy or focal loss, needs to be chosen to optimize the model for the ship detection task. The training data set needs to be fed into the model in batches during training. The data set should be shuffled before each epoch to prevent the model from memorizing the order of the data set. During training, the model computes the loss on the training data and backpropagates the error to adjust the model parameters. This is done using optimization algorithms such as stochastic gradient descent (SGD),To prevent overfitting, early stopping can be used to monitor the validation loss and stop the training when the validation loss stops improving.

The **Model Evaluation module** is essential in assessing the performance and reliability of the trained prediction model. This module employs evaluation metrics such as accuracy, precision, recall, and F1 score to measure how well the model performs in predicting whether the image contain ship or not. By comparing the model's predictions to the ground truth data, the module helps determine the model's effectiveness and identifies areas for improvement. The evaluation results provide insights into the model's strengths and weaknesses, allowing for adjustments or fine-tuning to enhance its performance and ensure its suitability for real-world applications.

The **Prediction module** is the core component of ship detection machine learning project. It utilizes the trained model to make predictions on new or unseen data. The new image need to undergo the same pre-processing steps that were applied to the training data, such as image normalization, resizing, and data augmentation. This ensures that the input images are in the same format as the training data and can be correctly processed by the model. he trained CNN or ResNet50 model is used to make predictions on the pre-processed images. The model computes the probability of each pixel or region of the image belonging to a ship or non-ship class.

5.2 System Architecture

A generic deep learning problem undergoes the following stages in development. Each stage can be distinct or sometimes overlapped with another stage. Fig 5.1 shows the various stages

involved in solving a generic deep learning problem.

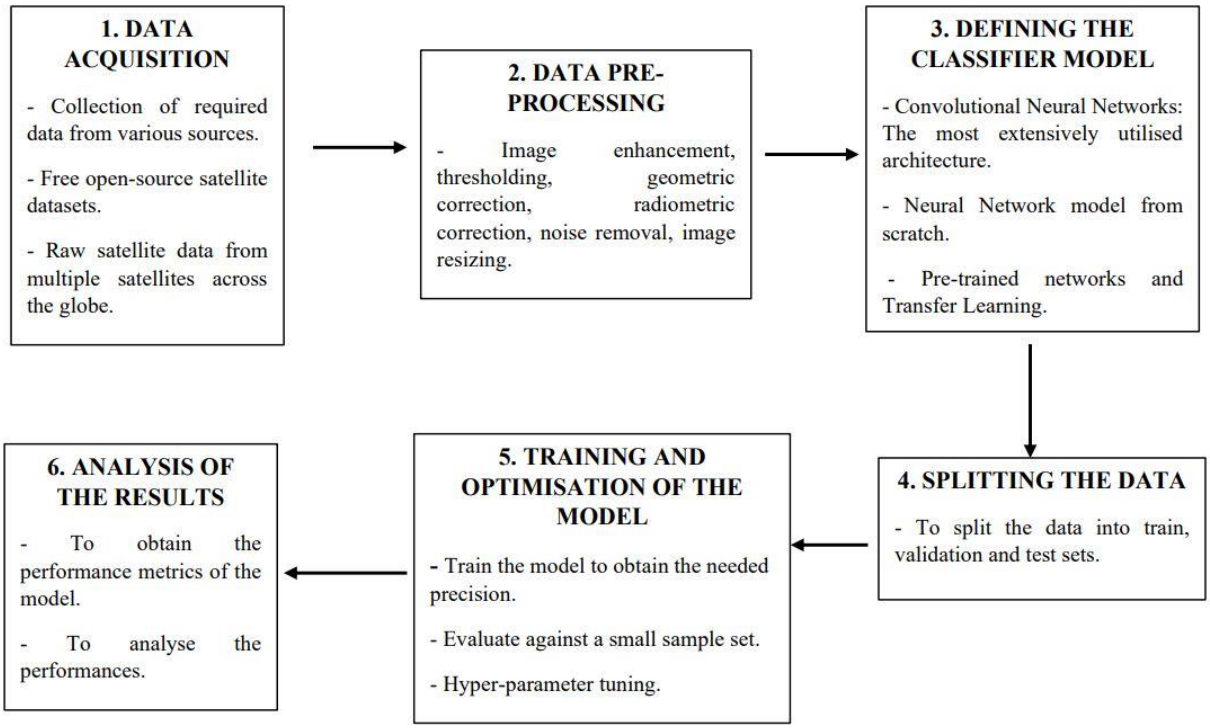


Fig 5.1: High Level Design

A more detailed view on each of the steps are given below.

1. DATA ACQUISITION.

A total of two datasets have been acquired from two different sources.

First, a dataset titled “Ships in satellite imagery” from Kaggle. This dataset consists of 4000 optical aerial images (1000 ship & 3000 no-ship). Fig 4.2 shows some images from this dataset. Hereafter referred to as Dataset A. Second, the MASATI-V2 dataset has been collected from a US based Military database. Hereafter referred to as Dataset B. The dataset consists of about 7000 images across 7 classes.

Table 5.1 gives a detailed view of the distribution of classes. Fig 5.3 shows some images from this dataset.

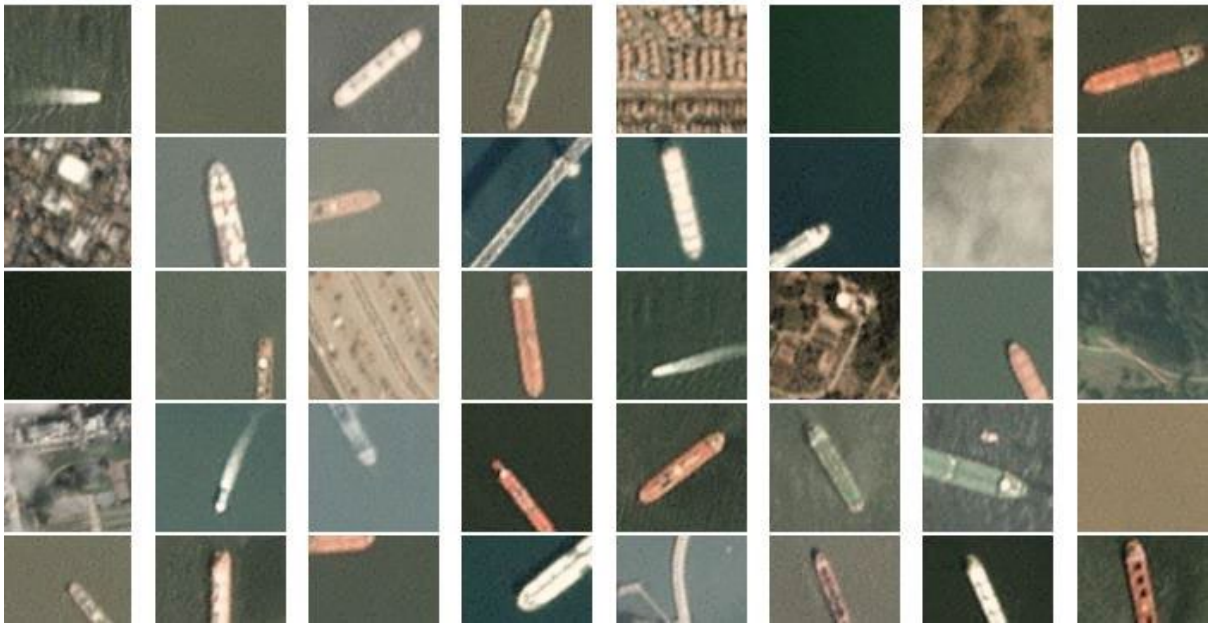


Fig 5.2: Some images in the dataset “Ships in Satellite Imagery”.

Fig 5.3: Some images in the MASATI dataset.

2. DATA PRE-PROCESSING.

Data pre-processing is a stage where in the acquired data is brought into the form of input as per our model requirements. It includes image resizing, noise removal etc. The images of dataset B have been resized to 224 * 224 pixels.

Class	# samples	Description
Ship	1,027	Sea with a ship (no coast)
Detail	1,789	Ship details
Coast & ship	1,037	Coast with ships
Multi	304	Multiple ships (with and without coast)
Sea	1,022	Sea (no ships)
Coast	1,132	Coast (no ships)
Land	1,078	Land (no sea)

Table 5.1: Distribution of classes in the MASATI v2 dataset

3. DEFINING THE CLASSIFIER MODEL.

The most commonly used algorithm for an image classifier is a deep learning model (CNN architecture).

Two CNN models have been developed; one a pre-defined CNN architecture (hereafter referred to as model 1) and another, a transfer learning model with the ResNet50 architecture as input layer (hereafter referred to as model 2).

4. SPLITTING THE DATA

The dataset has been split on the train and test data in the ratio 0.75:0.25. Further, the train data has been split into train and validation data in the ratio 0.80:0.20.

4.1 DATA AUGMENTATION

Further, to increase the amount of data for training, data augmentation has been used. Data augmentation is a process where in new data can be generated from the existing data by making some small changes to it.

5. TRAINING AND OPTIMIZATION OF THE MODEL.

The models have been trained for 10 epochs each and the results are compared. Initially, the weights of the ResNet50 layer have been freezed and trained.

Next, as a step towards finetuning, the residual layers and activation layers have been trained

and another model has been developed. This finetuned model is hereafter referred to as Model 3.

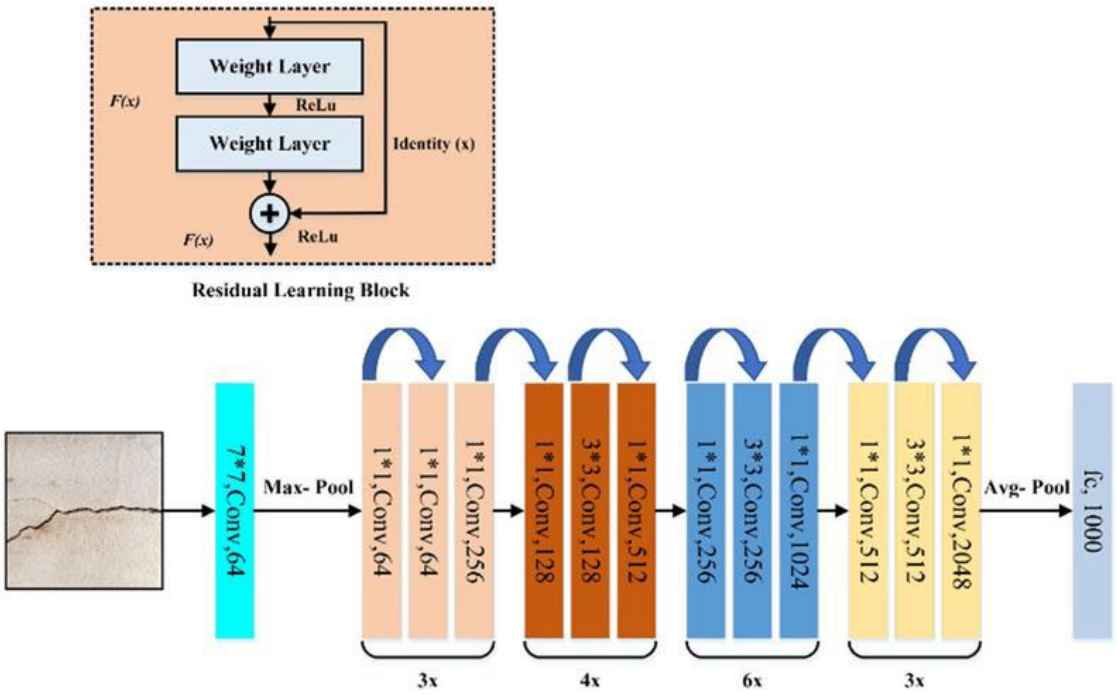


Fig 5.4: ResNet-50 Architecture

6. ANALYSIS OF RESULTS.

The results obtained from all the 3 models have to be obtained and compared using different performance metrics like accuracy and loss graphs, ROC curve, Confusion matrix etc.,

5.3 Data flow Diagram

5.3.1 Level 1: Data Collection and Preprocessing

- Collecting a large number of images, including both ship and no-ship.
- Preprocessing the images to ensure they are of uniform size and resolution, and removingany noise or artifacts.

- Labeling the images based on their class.

Fig 5.5: Data Collection and preprocessing

5.3.2 Level 2: Feature Extraction using ResNet

- Using a pre-trained ResNet model to extract high-level features from the images.
- Removing the fully connected layers of the ResNet model to obtain a feature extractor.
- Splitting the feature set into training and testing subsets.

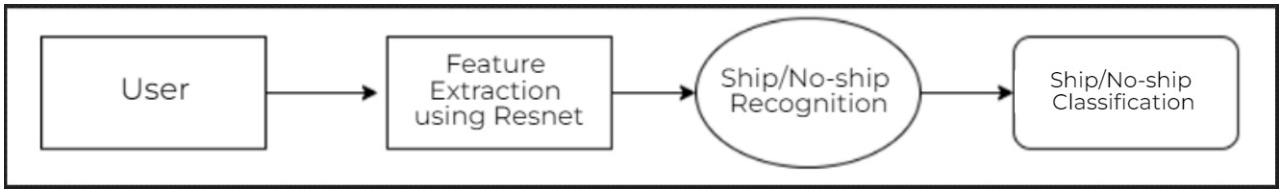


Fig 5.6 : Feature Extraction using ResNet

5.3.3 Level 3: Model Training

- Training the selected model on the training subset of ResNet features as it was having thehighest accuracy among all.
- Tuning the model hyperparameters to optimize its performance.

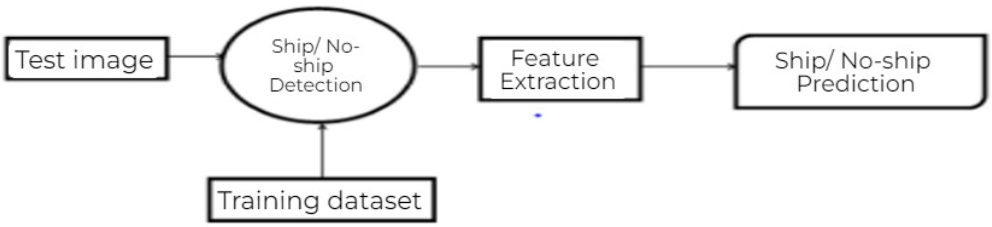


Fig 5.7: Model Training

5.3.4 Level 4: Model Evaluation and Testing

- Evaluating the model using accuracy as one of the parameters .
- Using confusion matrices and ROC curves to visualize the model's performance.
- Refining the model if necessary based on the evaluation results.

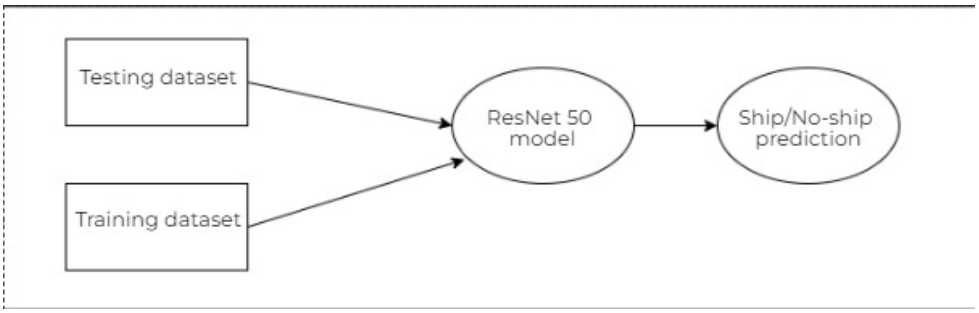


Fig 5.8: Model Evaluation and Testing

5.4 Sequence Diagram

The sequence diagram for the machine learning prediction model showcases the step-by-step process involved in the project. First the user uploads the dataset then it will be pre-processed to remove used. Then data is split into training and validation data which will be given for the models to train. Trained models will be saved and used to predict the new unseen images whether it contains ship or not and the output will be sent to the user

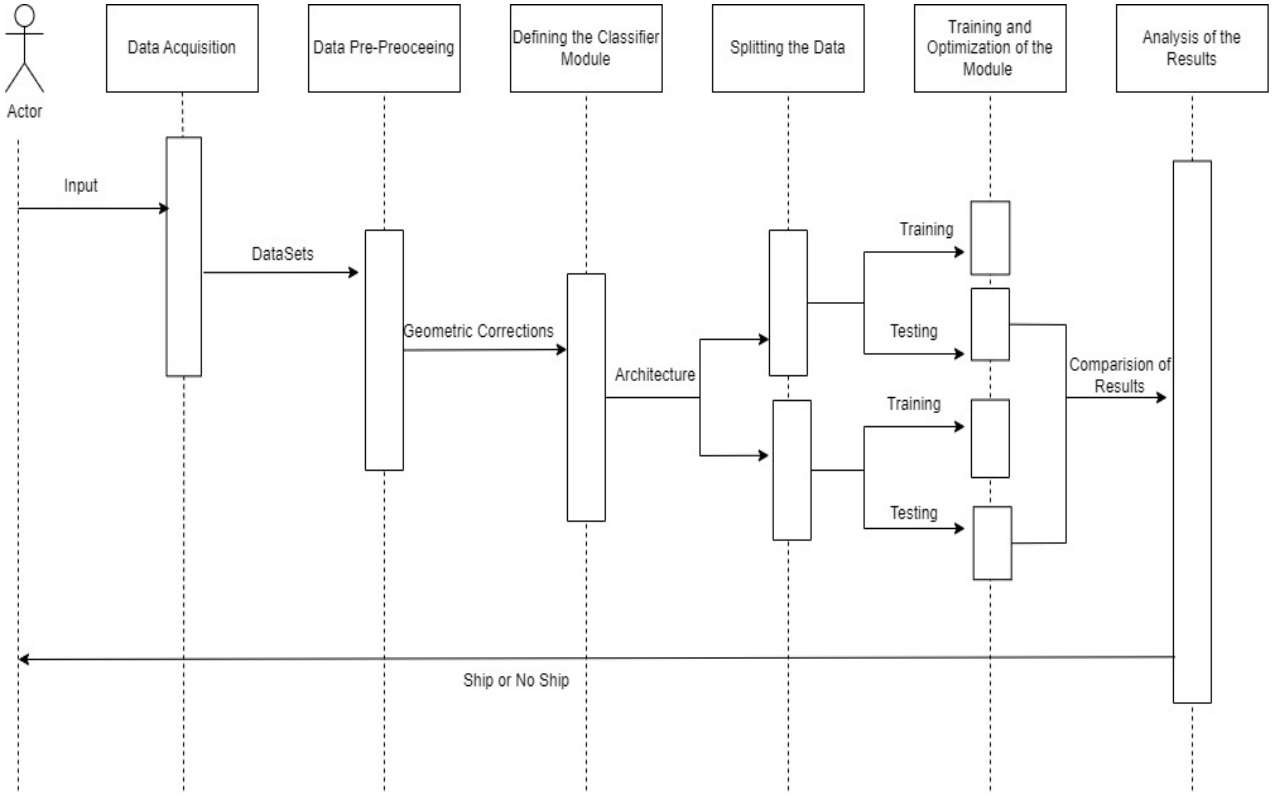


Fig 5.9: Sequence diagram for Ship Detection

5.3 Activity Diagram

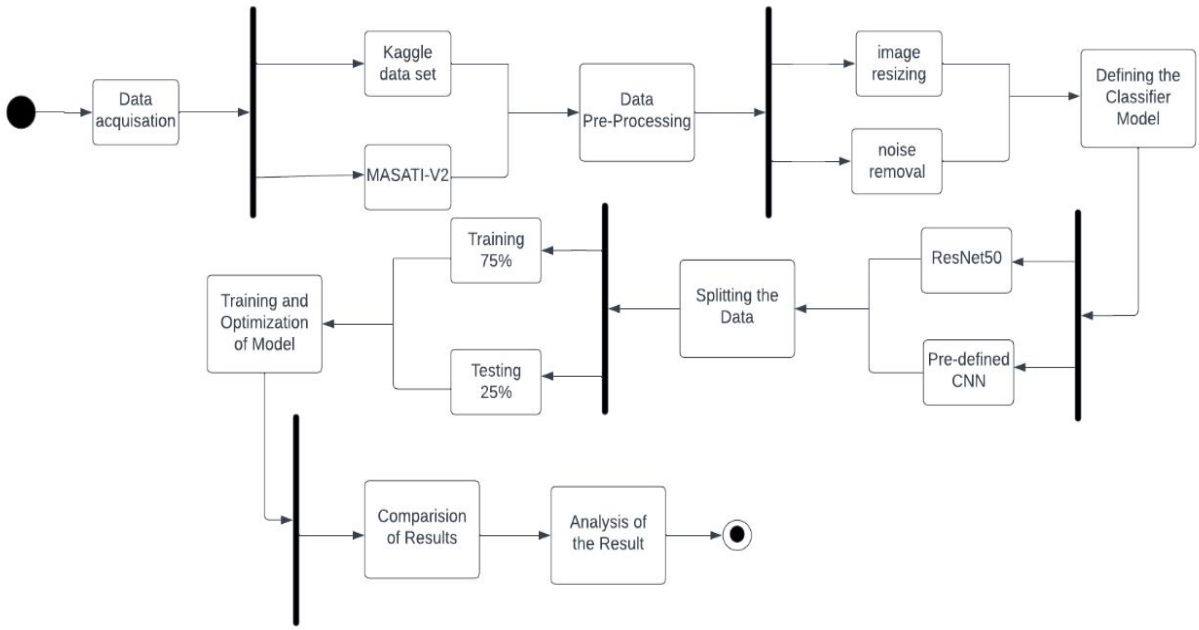


Fig 5.10: Activity diagram for Ship Detection

CHAPTER 6
IMPLEMENTATION

CHAPTER – 6

IMPLEMENTATION

6.1 Concept

In the project of detecting ship or not from optical images using machine learning techniques, deep learning and the ResNet50 architecture play a crucial role. Deep learning enables the automatic extraction of complex features and patterns from the optical images, improving the accuracy of ship detection. ResNet50, with its ability to handle deep networks effectively, enhances the performance of ship detection models by capturing intricate ship patterns in the images. By leveraging the residual connections, ResNet50 mitigates the challenges of training deep models and facilitates the accurate classification of images into ship or no-ship categories. The combination of deep learning (CNN) and ResNet50 in this project enables the development of a robust and accurate detection system for marine security.

● Firstly, we collected dataset from Kaggle and Masati-v2 and performed shuffling of the ship and no ship images.

● Applying the pre-processing techniques to enhance the quality and consistency of the ship images.

● Data augmentation is performed to increase the number of images by changing parameters like rotation_range, zoom_range and shear_range etc.

● Extracting some of the relevant features from the pre-processed images, such as colour histograms, texture descriptors, or deep learning-based features, to capture ship patterns.

●CNN model and ResNet50 models are defined according to our requirements.

● Train the selected model using the labelled ship and no-ship images and their corresponding extracted features, optimizing the model's parameters through techniques like gradient descent or backpropagation.

● Assessing the performance of the trained model using evaluation metrics like accuracy using techniques such as cross-validation to ensure robustness and generalization.

● Deploying the trained model in a user-friendly interface allowing the users to upload new images for predictions or probability scores for two different classes.

6.2 Algorithm

6.2.1 CNN model

Ship detection in satellite imagery is a crucial task with numerous applications in maritime surveillance, environmental monitoring, and security. Convolutional Neural Networks (CNNs) have emerged as a powerful tool for automated ship detection, revolutionizing the field with their ability to learn and extract complex features from images.

CNNs are deep learning models that mimic the human visual system. They consist of multiple layers of interconnected neurons, with each layer responsible for extracting specific features from the input data. In the context of ship detection, CNNs are trained to recognize patterns and shapes associated with ships, allowing them to accurately identify ships in satellite imagery.

To train a ship detection CNN, a large dataset of annotated satellite images is required. These images are labeled to indicate the presence or absence of ships. The CNN learns from these labeled examples, adjusting its internal parameters through a process called backpropagation to minimize the difference between its predicted outputs and the true labels. As the training progresses, the CNN becomes increasingly adept at recognizing ships in unseen images.

Once trained, the CNN can be applied to new satellite imagery for ship detection. The input image is fed into the CNN, which processes it through its layers, gradually extracting features at different levels of abstraction. The final layer produces a probability map indicating the likelihood of ship presence at each location. By applying a suitable threshold, ship candidates can be identified and extracted from the image.

The use of CNNs in ship detection has significantly improved the accuracy and efficiency of this task. It enables the analysis of large amounts of satellite imagery, facilitating timely and effective monitoring of maritime activities. As technology advances and datasets grow, CNN-based ship detection systems will continue to evolve, enhancing their capabilities and contributing to various maritime applications.

6.2.2 ResNet50 Model

ResNet50 is a convolutional neural network (CNN) architecture that incorporates residual connections. These connections allow the network to learn residual functions, enabling the training of very deep networks without encountering the vanishing gradient problem. The model consists of 50 layers with skip connections, which facilitate the flow of information across different layers.

In the context of ship detection in satellite imagery, ResNet50 is trained using large-scale datasets containing annotated images of ships. The model learns to recognize distinctive visual patterns that characterize ships, such as their shape, texture, and spatial relationships with the surrounding environment. It can identify ships of various sizes, types, and orientations, even in challenging conditions such as low resolution, occlusion, and varying lighting conditions.

To apply ResNet50 for ship detection, the satellite imagery is divided into smaller patches, and each patch is fed into the network for classification. The output of the model provides a probability score indicating the presence of a ship in the patch. By sliding the model across the entire image, ship candidates can be identified based on the regions with high probability scores.

Once ship candidates are detected, post-processing techniques like non-maximum suppression or clustering algorithms can be applied to refine the results and eliminate duplicate or false positive detections.

Overall, the ResNet50 model has demonstrated remarkable performance in ship detection tasks in satellite imagery, providing accurate and reliable results. Its ability to capture intricate visual features and its depth allow it to learn complex ship patterns effectively

6.3 Functional Modules

6.3.1 Collecting the dataset

This dataset is taken from the Masati-v2 dataset. This dataset consists of about 7.3K images of ship and no-ship images which is categorized into 7 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure. A new

directory containing test images is created later for prediction purpose.

```
path = "MASATI/SHIP1/"
dirs = os.listdir( path )
dirs.sort()
x_train=[]
```

```
def load_dataset():
    # Append images to a list •
    for item in dirs:
        if os.path.isfile(path+item):
            im = Image.open(path+item).convert("RGB")
            im = im.resize((224,224),0)
            im = np.array(im)
            x_train.append(im)
```

```
[ ] if __name__ == "__main__":

        load_dataset()

        # Convert and save the list of images in '.npy' format
        imgset=np.array(x_train)
        print(imgset.shape)
        #np.save("imgds.npy",imgset)
```

```
(2354, 224, 224, 3)
```

6.3.2 Data Preparation

```
[ ] x,y = shuffle(imgset,Y, random_state=2)
    # Split the dataset
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)
    del x,y,Y,imgset #doing this to free up RAM
```

```
[ ] print(len(x_train))
    print(len(x_test))
```

```
3052
763
```

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
        rotation_range=20,
        zoom_range=0.15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.15,
        horizontal_flip=True,
        fill_mode="nearest")
```

```
[ ] test_datagen = ImageDataGenerator()
```

```
[ ] batch_size = 8
    train_generator = train_datagen.flow(trainX, trainY, batch_size=batch_size)
    valid_generator = train_datagen.flow(trainX, trainY, batch_size=batch_size)
    test_generator = test_datagen.flow(x_test, batch_size=1,save_to_dir='./ships_kaggle/shipsnet/shipsnet/test_images')
```

After uploading dataset images, images are shuffled and split into 3 categories: training,validation and testing and data us augmented using ImageDataGenerator(). Then the

train,valid and test generators are created. The test images are saved for later prediction purpose.

6.3.3 Training the models

Model 1 - CNN

```
def prepare_model():
    model = Sequential()
    #model.add(restnet)
    #input_shape=(80,80,3)
    model.add(Conv2D(32,kernel_size=(2,2),activation='relu',input_shape=(224, 224, 3)))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Flatten())
    model.add(Dense(16, activation='relu'))
    model.add(Dense(2, activation='softmax'))
    model.layers[0].trainable=False
    model.compile(loss="binary_crossentropy",optimizer="adam",metrics=['accuracy'])
    return model
```

```
history=model.fit_generator(train_generator, validation_data=(valX, valY),
 steps_per_epoch=len(trainX) // batch_size,
 epochs=5,)
model.save("MASATI/CNN.h5")
```

```
Epoch 1/5
<ipython-input-43-a3bd133d243b>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  history=model.fit_generator(train_generator, validation_data=(valX, valY),
305/305 [==============================] - 49s 159ms/step - loss: 18.6190 - accuracy: 0.6145 - val_loss: 0.6831 - val_accuracy: 0.6088
Epoch 2/5
305/305 [==============================] - 48s 156ms/step - loss: 0.6765 - accuracy: 0.6153 - val_loss: 0.6762 - val_accuracy: 0.6088
Epoch 3/5
305/305 [==============================] - 48s 157ms/step - loss: 4.5163 - accuracy: 0.6161 - val_loss: 0.6707 - val_accuracy: 0.6105
Epoch 4/5
305/305 [==============================] - 47s 154ms/step - loss: 0.6681 - accuracy: 0.6161 - val_loss: 0.6694 - val_accuracy: 0.6105
Epoch 5/5
305/305 [==============================] - 48s 157ms/step - loss: 0.6671 - accuracy: 0.6157 - val_loss: 0.6688 - val_accuracy: 0.6105
```

As, we are considering a balanced dataset for our project. So, accuracy is the parameter which will help us to compare between the all 3 models. Model is trained using model.fit_generator() function passing the train generator obtained earlier and the validation data. Model is saved for future prediction purpose.

Model 2 and 3 – ResNet model with fine tuning

```
restnet = ResNet50(include_top=False, weights='imagenet', input_shape=(224,224,3),backend=tf.keras.backend, layers=tf.keras.layers, models=tf.keras.models, utils=tf.keras.utils)
output = restnet.layers[-1].output
output = keras.layers.Flatten()(output)
restnet = Model(restnet.input, outputs=output)
for layer in restnet.layers:
    layer.trainable = False
restnet.summary()
```

```
[ ]  #setting parameters to be finetuned
     from tensorflow import keras
     from keras import optimizers
     restnet.trainable = True
     set_trainable = False
     for layer in restnet.layers:
         if layer.name in ['res5c_branch2b', 'res5c_branch2c', 'activation_97']:
             set_trainable = True
         if set_trainable:
             layer.trainable = True
         else:
             layer.trainable = False
     layers = [(layer, layer.name, layer.trainable) for layer in restnet.layers]
     pd.DataFrame(layers, columns=['Layer Type', 'Layer Name', 'Layer Trainable'])
```

Here parameters are finetuned according to our requirement and model is trained in the similar manner as model 1 and model is saved.

```
#validation
score = model.evaluate_generator(valid_generator)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
<ipython-input-100-b5434ed1bd0c>:2: UserWarning: `Model.evaluate_generator` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports gener
  score = model.evaluate_generator(valid_generator)
Test loss: 0.5215266346931458
Test accuracy: 0.9381400942802429
```

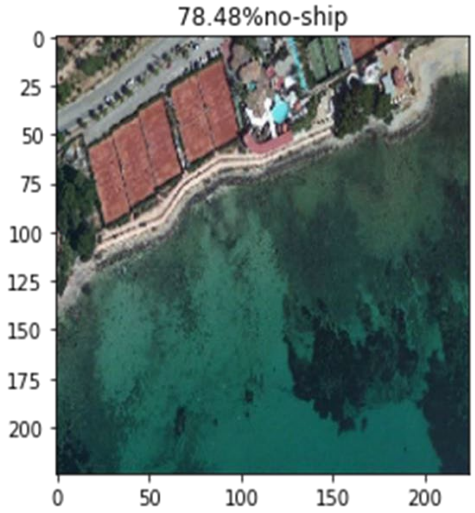
Model is validated using model.evaluate_genetaor() function passing the valid_generator obtained above.

6.3.4 Testing the model

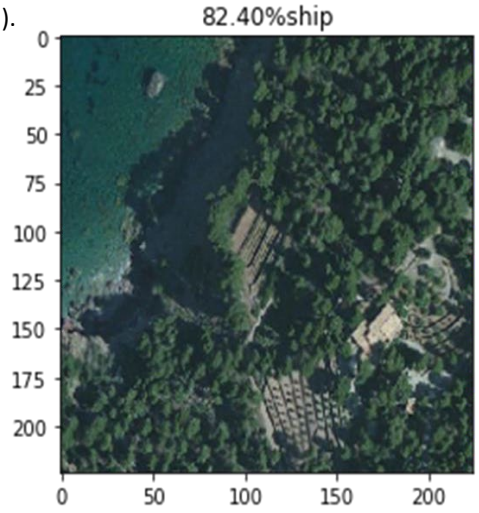
```
model=load_model("MASATI/RESNET_fine.h5")
y_pred = model.predict(x_test, batch_size=8)
#view first 30 predictions
prediction=y_pred[0:30]
for index, probability in enumerate(prediction):
  if probability[1] > 0.5:
        plt.title('%.2f' % (probability[1]*100) + '%no-ship')
  else:
        plt.title('%.2f' % ((1-probability[1])*100) + '%Ship')
  img=x_test[index]
  img = np.array(img/np.amax(img)*255, np.int32)
  plt.imshow(img)
  plt.show()
```

The predictions from the models is obtained using above defined function. Predictions is saved to ‘y_pred’ variable . Index and probability of the images are obtained from y_pred variable and predictions are shown with percentage of confidence whether the image contain image or no-ship.

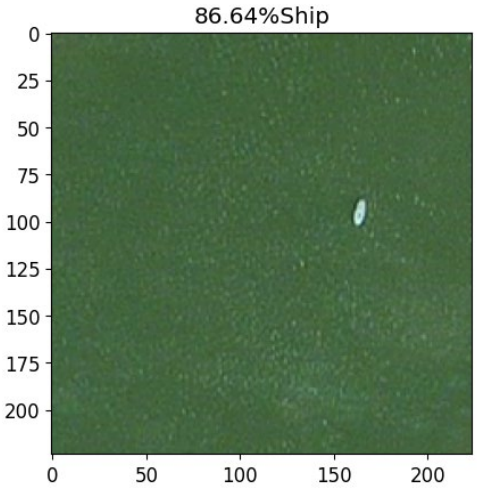
Model 1 - CNN

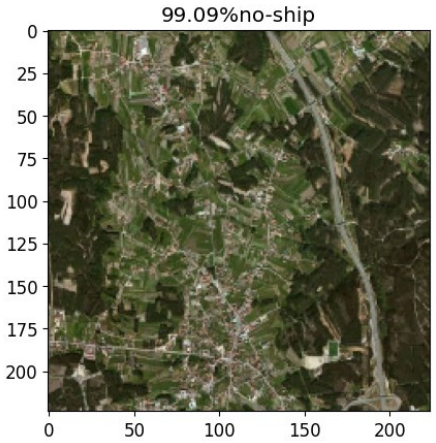


).

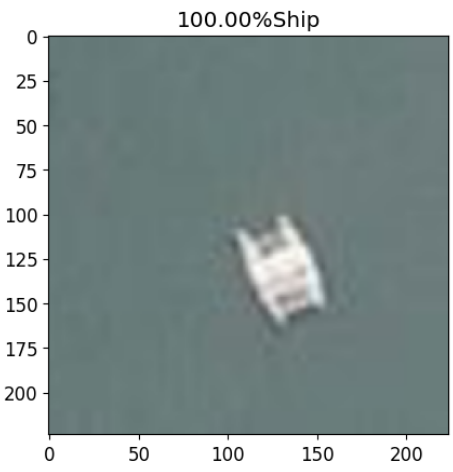


Model 2 – ResNet50





Model 3 – ResNet50 with fine tuning



CHAPTER 7
TESTING

CHAPTER – 7

TESTING

7.1 Methods of Testing

Testing methods are crucial in ensuring the quality and reliability of a system. Common testing methods include unit testing, where individual components are tested in isolation, integration testing, which verifies the interaction between different components, and system testing, which evaluates the system as a whole. Additionally, there are performance testing, security testing, and user acceptance testing, which assess the system's performance, security measures, and user satisfaction, respectively. These testing methods help identify and rectify any errors, vulnerabilities, or performance issues, ensuring that the system meets the required standards and specifications.

7.1.1 Unit Testing

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses on the verification efforts of the smallest unit of software designed in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during the programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example, the validation check is done for varying the user input given by the user with the validity of the data entered. It is very easy to find error debut the system.

Table 7.1.1: Unit Testing Table

Function Name	**Test Results**
Uploading Image	Tested for uploading different types and sizes of images.
Detecting class	Tested for different images of optical images which includedboth ship and non-ship.
Get Prediction	Tested if the output is displayed successfully.

7.1.2 System Testing

System testing is a type of software testing that involves testing the entire system or software application as a whole to verify that it meets the specified requirements and works correctly. It is done after unit testing and integration testing and focuses on testing the system's functionality, performance, security, usability, and compatibility. System testing is important to ensure that the system is ready for release and works as expected in different environments and under different conditions.

Table 7.1.2: System Testing Table

Function Name	**Test Results**
Functionality checking	Verifying that the system functions as expected forthe detection of ship.
understandable for users	Identifying the areas of improvement and make necessarychanges to the system
Performance	Feed a large number of ship images to the system to test itsperformance under expected and peak loads.
Compatibility	Checking the system works as expected across differentplatforms and environments.

7.1.3 Functional Testing

Functional testing is a type of software testing that checks whether a system or software applicationis functioning correctly as per the specified requirements. It involves testing the system's functionality, features, and user interface to ensure that they work as expected. The goal of functional testing is to identify defects or issues in the system and verify that it meets the user's needs and expectations.

Table 7.1.3: Functional Testing Table

Function Name	**Test Results**
Input Image Testing	Provided input images of ship and non-ship to the systemfor verifying that the system correctly identifies and classifies the images as ship or no-ship.
System correctly classifies	Test the system's ability to classify images with multiple ships.

Error Handling	Providing input images of ship with unexpectedformats or content to test the system.

7.1.4 Integration Testing

Integration testing is a type of software testing that focuses on testing the interfaces between different software components or modules to ensure that they work together correctly as a whole.It involves combining individual modules or components and testing them as a group to verify that they integrate seamlessly and meet the specified requirements. The goal of integration testing is toidentify defects or issues that may arise from the interaction between the different components andto ensure that the system works correctly as a whole. The test cases focus on testing data communication, control flow, error handling, and system performance under different integrationscenarios. Integration testing is usually done after unit testing and before system testing to ensurethat the system works well as a complete and integrated unit.

7.1.5 User Acceptance Testing

User acceptance testing is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer for their satisfaction, and checks whether the application is working according to given business scenarios, real-time scenarios.

In our project we trained our model using some images and then it was tested with over 100 images. The accuracy using the existing labeled classes was found to be around 94%

7.2 Test Cases

Table 7.2: Test Cases Table

Test case:

Test #	Test Data(input)	Expected Result	Actual Result	Pass/ Fail
1	High quality satellite image	Classification as ship (or) Classification as no ship	Classification as ship(or) Classification as no ship	Pass
2	High quality satellite image	Classification as ship	Classification as ship	Pass
3	Low quality satellite image	Classification as ship	Classification as ship	Pass

CHAPTER 8

PERFORMANCE ANALYSIS

CHAPTER – 8

PERFORMANCE ANALYSIS

Multiple Performance metrics were calculated for all the 3 models.

8.1Accuracy and Loss Graphs:

Accuracy and loss graphs are visual representations of a machine learning model's performance during training. The accuracy graph displays the percentage of correct predictions made by the model during training. This can help identify when the model has reached its optimal accuracy and if further training is necessary. The loss graph shows the difference between the predicted and actual values, which is also known as the cost function. As the model learns, the cost function decreases, indicating that the model is getting closer to making accurate predictions. However, if the cost function does not decrease over time, it may indicate that the model is not learning as expected, and adjustments may need to be made to the model architecture or training process. By analyzing the accuracy and loss graphs, developers can make informed decisions about the model's architecture, training parameters, and when to stop training the model to achieve the best possible performance.

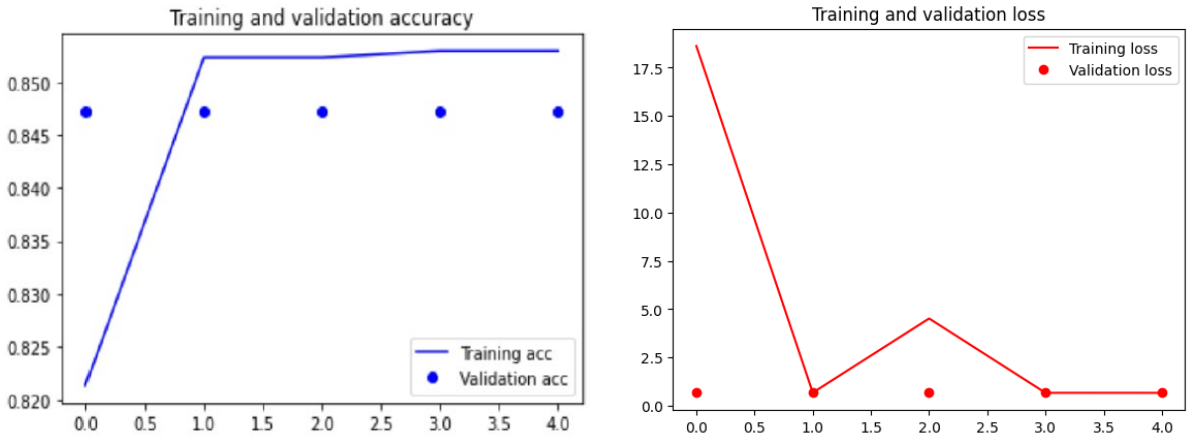


Fig 8.1: Accuracy and Loss Graph for model 1

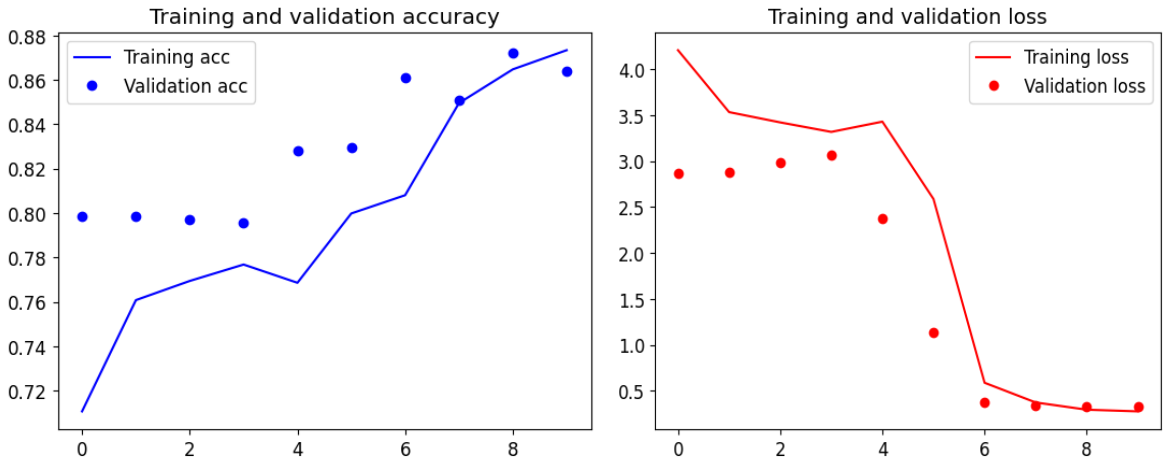


Fig 8.2: Accuracy and Loss Graph for model 2

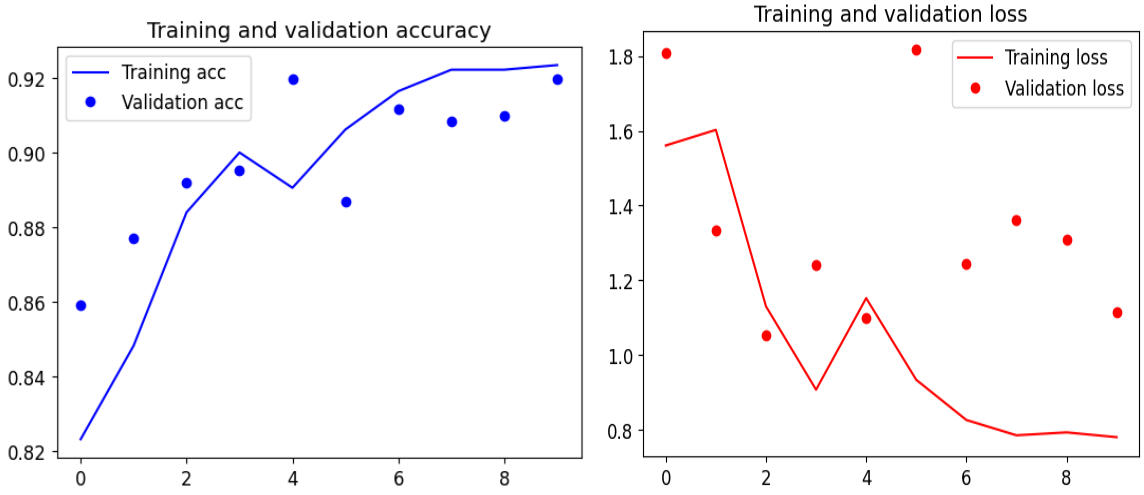


Fig 8.3: Accuracy and Loss Graph for model 3

8.2 Confusion matrix

A normalized confusion matrix is a table that visualizes the performance of a classification model by comparing the predicted labels with the actual labels of the test data. It shows the number of true positive, true negative, false positive, and false negative predictions, normalized by the total number of instances in each class. Normalizing the matrix helps to compare the performance of different models on datasets with different class distributions. It also provides additional information, such as precision, recall, and F1-score, which can be used to evaluate the overall performance of the model. By analyzing the normalized confusion matrix, developers can identify which classes are being predicted accurately and which ones need improvement, allowing them to optimize the model's performance.

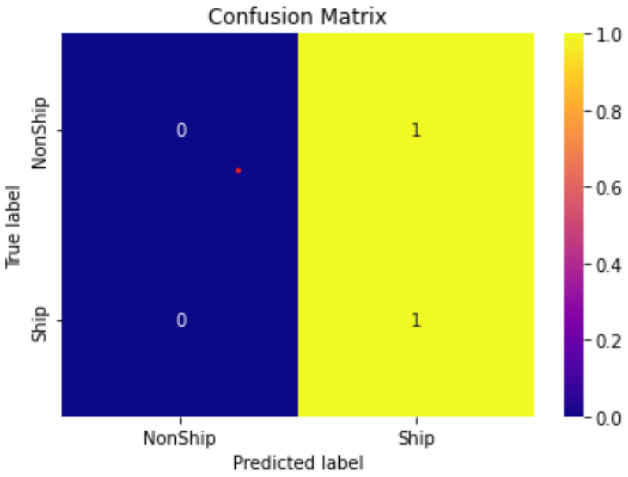


Fig 8.4 : Confusion Matrix for Model 1

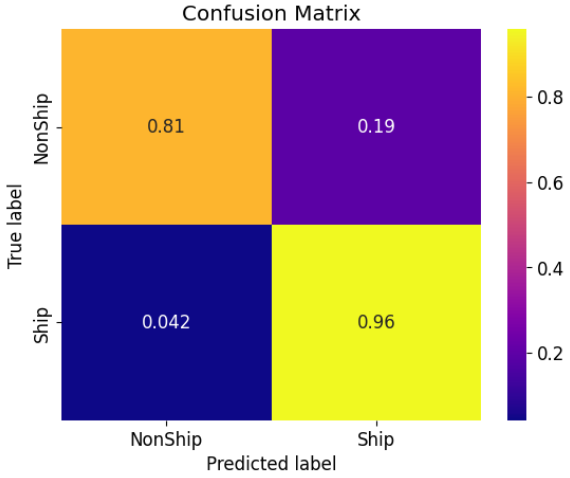


Fig 8.5 : Confusion Matrix for Model 2

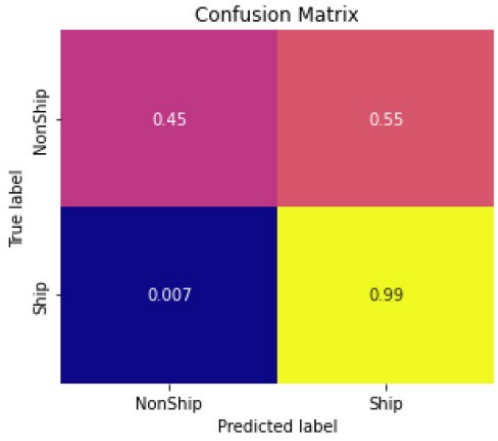


Fig 8.6 : Confusion Matrix for Model 3

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER – 9

CONCLUSION AND FUTURE ENHANCEMENT

Ship detection plays a vital role in the maritime security. Hence, a classifier model is developed to classify a given input image as “ship” and “no-ship”. From the above experimental results, it is conclusive that the model 3 (model in which the pre-trained ResNet50 was finetuned) is giving the best results, i.e., 82.6% and 93 % accuracy in both the datasets respectively. As a scope of future work, there are 2 things:

- Create a dataset which is more suitable for the Indian coastline from the Indian satellite data.
- Dive into multi-class classification of the identified ships.

The application of deep learning techniques for the detection of ships in satellite imagery has proven to be highly effective and promising. This final project report has highlighted the key findings and outcomes of the study, demonstrating the potential of deep learning in enhancing ship detection capabilities. Through the utilization of convolutional neural networks (CNNs) and other deep learning architectures, significant advancements have been achieved in accurately identifying ships within satellite imagery. The development of robust models, such as Faster R-CNN, YOLO, or SSD, has allowed for efficient object detection and localization, enabling automated ship detection with high precision and recall rates.

The advantages of deep learning techniques lie in their ability to learn complex patterns and features directly from raw satellite imagery, eliminating the need for manual feature engineering. By training these models on large-scale datasets, the performance of ship detection algorithms has been significantly improved, surpassing traditional methods in terms of accuracy and efficiency. Furthermore, transfer learning has emerged as a valuable approach for ship detection, enabling the transfer of knowledge from pre-trained models on large-scale datasets to smaller ship-specific datasets. This technique has demonstrated its ability to achieve good performance even with limited training data, reducing the need for extensive and time-consuming data annotation.

However, some challenges still exist in ship detection using deep learning techniques. These challenges include handling variations in ship appearance due to different imaging conditions, such as weather, lighting, and viewpoint changes. Additionally, the presence of complex backgrounds and occlusions can pose difficulties for accurate ship detection. Future research in this field should focus on addressing these challenges and further improving the performance of ship detection models. This can be achieved through the development of more robust deep learning architectures, the exploration of novel data augmentation techniques, and the integration of multi-modal data sources for enhanced ship detection accuracy. Overall, the application of deep learning techniques for ship detection in satellite imagery holds great promise for various domains, including maritime security, environmental monitoring, and maritime traffic management. The findings of this project underscore the significant potential of deep learning in advancing ship detection capabilities and contribute to the ongoing development of more reliable and efficient systems for ship detection in satellite imagery.

The project report on the detection of ships in satellite imagery using deep learning techniques presents several opportunities for future enhancements. Here are some potential areas for further improvement and research:

1. **Handling Variations in Ship Appearance**: Ships can appear differently in satellite imagery due to various factors such as weather conditions, lighting variations, and different viewpoints. Future research can focus on developing robust models that can handle these appearance variations effectively. Techniques such as domain adaptation and data augmentation can be explored to improve the generalization capabilities of ship detection models.

2**. Detection of Small and Partially Occluded Ships**: Deep learning models may struggle with detecting small-sized ships or ships that are partially occluded by other objects or the environment. Enhancements can be made to the architectures and training strategies to improve the detection of such challenging instances. Techniques like anchor-free detectors and attention mechanisms can be incorporated to handle small and occluded ships more effectively.

3. **Integration of Multi-Modal Data**: Satellite imagery can be enriched by incorporating other modalities such as radar data or Automatic Identification System (AIS) data. Combining multiple data sources can provide complementary information for ship detection, enhancing

the accuracy and reliability of the detection models. Future research can explore methods for effectively fusing multi-modal data and leveraging their synergistic benefits.

4**.Real-Time Ship Detection**: Real-time ship detection is crucial for time-critical applications, such as maritime surveillance or emergency response. Future work can focus on optimizing deep learning models for real-time inference without sacrificing detection performance. Techniques like model compression, network pruning, and hardware acceleration can be explored to achieve efficient and fast ship detection in real-time scenarios.

5**. Data Collection and Annotation**: The availability of large-scale, annotated ship datasets is crucial for training accurate deep learning models. Future efforts can be directed towards creating more diverse and comprehensive datasets with accurately labeled ship instances. Crowd-sourcing techniques, active learning strategies, and data synthesis approaches can be explored to facilitate the collection and annotation of ship data efficiently.

6**. Transfer Learning for Limited Data**: Transfer learning has proven to be effective for ship detection with limited labeled training data. Further research can investigate techniques for better leveraging pre-trained models and transfer learning in scenarios with extreme data scarcity. Techniques like self-supervised learning and semi-supervised learning can be explored to improve the performance of ship detection models with limited labeled data.

7. **Model Interpretability and Explainability**: Deep learning models often lack the ability interpretability, making it challenging to understand the reasoning behind their predictions. Future research can focus on developing techniques to interpret and explain the decisions made by ship detection models. This can help build trust and understanding, especially in applications where decision-making has significant implications.

By addressing these future enhancements, the field of ship detection in satellite imagery using deep learning techniques can continue to advance, leading to more accurate, efficient, and reliable systems with broader applicability in maritime-related domains.

BIBLIOGRAPHY

[1] Jiang, Jiahuan, et al. “High-speed lightweight ship detection algorithm based on YOLO-v4 for three-channels RGB SAR image.” Remote Sensing 13.10 (2021): 1909.

[2] Rana Muhammad Usman, Junhua Yan, Imran Qureshi, “Research on Ship Detection and Classification Using Deep Learning Approach”, International Journal of Mod- ern Research in Engineering and Technology (IJMRET), December 2021.

[3] Muhammad Kabir Dauda, Abubakar Salihu Abbu, Naziru Saleh, Kabiru Ibrahim Musa, Hussaini Muhammad Khamis, Abubakar Umar, “Transfer Learning Strategy For Satellite Image Classification Using Deep Convolutional Neural Network”, International Journal of Advanced Engineering and Management Research,2020.

[4] Richa, J. “Smart ship detection using transfer learning with ResNet.” 2019 International Research Journal of Engineering and Technology (IRJET), India (2019).

[5] Matasci, Giona, et al. “DEEP LEARNING FOR VES- SEL DETECTION AND IDENTIFICATION FROM SPACEBORNE OPTICAL IMAGERY.” ISPRS Annals of Photogrammetry, Remote Sensing Spatial Information Sciences 3 (2021).

[6] Li, Yifan, et al. “Machine Learning Methods for Ship Detection in Satellite Images.” write down Germany, IEEE (2013).

[7] Marzuraikah Mohd, Stofa, Siti Zulaikha Muhammad Zaki, Mohd Asyraf Zulkifle. “A deep learning ap- proach to ship detection using satellite imagery.” IOP Conference Series: Earth and Environmental Science. Vol. 540. No. 1. IOP Publishing, 2020.

[8] Kumar, Durga, and Xiaoling Zhang. “Ship Detection Based on Faster R-CNN in SAR Imagery by Anchor Box Optimization.” 2019 International Conference on Control, Automation and Information Sciences (ICCAIS). IEEE, 2019.

[9] Mahmud, M. P., Tahir, A., Munawar, H. S., Kouzani, A. Z., Akram, J., Adil, M., Ali, S., (2022). “Automatic target detection from satellite imagery using machine learning. Sensors”, 22(3), 1147.

[10] Wu, Fei, et al. “Inshore ship detection based on convolutional neural network in optical satellite images.” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 11.11 (2018): 4005-4015.

[11] Bereta, Konstantina, Dimitris Zissis, and Raffaele Grasso. “Automatic maritime object detection using satellite imagery.” Global Oceans 2020: Singapore–US Gulf Coast. IEEE, 2020.

[12] Z. Baijun, Z. Yongsheng, L. Ting and Y. Shun, “Ship Detection Algorithm based on Improved YOLO V5,” 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE), 2021.
[13] P. Zheng, X. Wu, Z. -Q. Zhao, S. -T. Xu and, “Object Detection with Deep Learning: A Review,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019.
[14] Gadamsetty, Samhitha, et al. “Hash-based deep learning approach for remote sensing satellite imagery detection.” Water 14.5 (2022): 707.
[15] Krestenitis, Marios, et al. “Oil spill identification from satellite images using deep neural networks.” Remote Sensing 11.15 (2019): 1762.
[16] E. Schwarz, F. Heymann, S. Voinov and R. Bill. “Multiclass Vessel Detection from High Resolution Optical Satellite Images Based on Deep Neural Networks,” IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019, pp. 166-169, doi: 10.1109/IGARSS.2019.8900506.
[17] Liu, Ying, et al. “Ship detection and classification on optical remote sensing images using deep learning.” ITM Web of Conferences. Vol. 12. EDP Sciences, 2017. [18] Yang, Xue, et al. “Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid net- works.” Remote Sensing 10.1 (2018): 132.
[19] O. Duman and M. Kartal. “Ship Detection from Optical Satellite Images with Deep Learning,” 2019 9th International Conference on Recent Advances in Space Technologies (RAST), 2019, pp. 479-484, doi: 10.1109/RAST.2019.8767844.
[20] Z. Hong et al., “Multi-Scale Ship Detection from SAR and Optical Imagery Via a More Accurate YOLOv3,” in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 6083- 6101, 2021.

APPENDIX

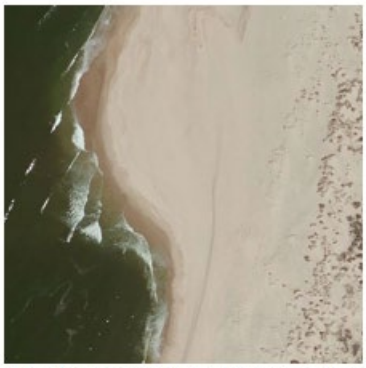
Appendix A: Screen Shots

Making Predictions foe a single image

```
#To Predict singlw image
import tensorflow as tf
from PIL import Image

# Load the saved model
model = tf.keras.models.load_model('/content/drive/MyDrive/MASATI/RESNET.h5')

# Load the image you want to classify
image = Image.open('/content/drive/MyDrive/MASATI/SHIP/d0003.png')

# Preprocess the image
image = image.resize((224, 224))  # Resize the image to match input size of the model
image.show()
image = tf.keras.preprocessing.image.img_to_array(image)  # Convert the image to a numpy array
image = tf.keras.applications.resnet50.preprocess_input(image)  # Preprocess the image to match the input requirements of the ResNet50 model

# Make a prediction
predictions = model.predict(image.reshape(1, 224, 224, 3))  # Reshape the image to match the input shape of the model and make a prediction
predicted_class = tf.argmax(predictions, axis=-1)  # Get the index of the class with the highest probability


# Interpret the output
if predicted_class == 0:
    print('The image is a ship')
else:
    print('The image is not a ship')
```

```
1/1 [==============================] - 2s 2s/step
The image is a ship
```

```
1/1 [==============================] - 1s 1s/step
The image is a ship
```



```
1/1 [==============================] - 1s 1s/step
The image is not a ship
```

```
1/1 [==============================] - 1s 1s/step
The image is not a ship
```

Appendix B: Abbreviation

ML – Machine Learning

Machine learning is a field devoted to understanding and building methods that let machines"learn" – that is, methods that leverage data to improve computer performance on some set of tasks

CNN - Convolutional Neural Network

In deep learning, a convolutional neural network is a class of artificial neural networks mostcommonly applied to analyze visual imagery.

DL – Deep Learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers.

ResNet – Residual Network

Residual Network (ResNet) is a deep learning model used for computer vision applications.

PAPER PUBLICATION DETAILS

JETIR (International Journal of Emerging Technologies and Innovative Research, An International Open Access Journal) explores advances in research pertaining to applied, theoretical and experimental Technological studies. The goal of JETIR is to promote scientific information interchange between researchers, developers, students, and practitioners working in and around the world.

Paper ID : JETIR2305019

JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Ref No : JETIR / Vol 10 / Issue 5 / 019

Confirmation Letter

To,
Mohith A N
Published in : Volume 10 | Issue 5 | 2023-05-04

ISSN 2349-5162

Subject: Publication of paper at International Journal of Emerging Technologies and Innovative Research .

Dear Author,

With Greetings we are informing you that your paper has been successfully published in the International Journal of Emerging Technologies and Innovative Research (ISSN: 2349-5162). Following are the details regarding the published paper.

About JETIR	: An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal Impact Factor Calculate by Google Scholar and Semantic Scholar | AI-Powered Research Tool, Multidisciplinary, Monthly, Multilanguage Journal Indexing in All Major Database & Metadata, Citation Generator, Impact Factor: 7.95, ISSN: 2349-5162
UGC Approval	: UGC and ISSN Approved - UGC Approved Journal No: 63975 | Link: https://www.ugc.ac.in/journallist/subjectwisejurnallist.aspx?tid=MjM0OTUxNjI =&&did=U2VhcmNoIGJ5IElTU04=
Registration ID	: JETIR_514472
Paper ID	: JETIR2305019
Title of Paper	: Detection Of Ships in Satellite Imagery Using Deep Learning Techniques
Impact Factor	: 7.95 (Calculate by Google Scholar)
DOI	:
Published in	: Volume 10 | Issue 5 | 2023-05-04
Publication Date	: 2023-05-04
Page No	: a142-a147
Published URL	: http://www.jetir.org/view?paper=JETIR2305019
Authors	: Mohith A N, Rehan Nadeemulla, Rahul Ghosh , Md Shahid, Dr. Sreenivasa B C

Thank you very much for publishing your article in JETIR. We would appreciate if you continue your support and keep sharing your knowledge by writing for our journal JETIR.

Indexing CiteSeer SSRN .docstoc Google Scribd. publons MENDELEY Semantic Scholar

Editor In Chief
International Journal of Emerging Technologies and Innovative Research
(ISSN: 2349-5162)

www.jetir.org | editor@jetir.org | Impact Factor: 7.95 (Calculate by Google Scholar)

An International Scholarly Open Access Journal, Peer-Reviewed, Refereed Journal

JETIR.ORG Email: editor@jetir.org