ABSTRACT

Ship identification in satellite images poses significant challenges within the domain of remote sensing. Its importance extends to critical areas such as security, encompassing concerns such as military attacks, accidents, illegal transportation of goods, illegal fishing, territorial violations, and ship hijackings. Additionally, effective traffic management and smuggling prevention heavily rely on accurate ship identification. While synthetic aperture radar (SAR) has historically dominated maritime monitoring, researchers are increasingly exploring optical satellite images as a potential alternative. Previous ship detection techniques have utilized Computer-based image processing vision methods. However, this study proposes a novel approach employing a convolutional neural network (CNN) based method to accurately identify ships in satellite data. The suggested approach entails the utilization and assessment of a custom-designed deep learning model based on CNN architecture to recognize ships in satellite photos The research contributes to the advancement of maritime surveillance by offering an innovative solution that complements traditional methods, aiding naval authorities in effectively monitoring and securing maritime boundaries.

Keywords - ship detection, optical images, convolutional neural networks.

1. INTRODUCTION

1.1 project description

Ship detection plays a important role for various maritime applications, including maritime security, vessel traffic management, environmental monitoring, and search and rescue operations. Throughout time, various technologies have emerged and evolved and been employed to detect and track ships, and have been used like the AIS (Automatic Identification System) represents a widely adopted solution.

However, AIS has its limitations, which necessitates the necessity for identifying ships utilizing pictures captured by satellites AIS is a radio-based system that utilizes transponders on ships toexchange information about their identity, position, course, and speed. It provides real- time data, enabling efficient vessel tracking, collision avoidance.

AIS is particularly valuable in coastal areas, and busy shipping lanes where vessels are equipped with AIS transponders. However, AIS coverage is limited to areas within range offshore-based or satellite-based receivers, leaving vast stretches of open seas and remote regions devoid of AIS information. Moreover, AIS signals can be easily blocked or manipulated, compromising its reliability in certain situations.

In contrast, satellite imagery provides a comprehensive and wide-range of the oceans, allowing for the detection and tracking of ships across large areas. Satellite images offer high-resolution visual Information that can be examined for identification of ships, even in remote or poorly covered regions. By leveraging advanced image processing and computer vision techniques, the utilization of satellite images for identifying ships can offer valuable understandings regarding to track vessel activities, routes, and behavior of the movement and surveillance of the ships moving in that area, avoid illegal activities

1.2 Tools and Technologies used

1.2.1 Python programming language

Python is a versatile programming language utilized for developing applications, websites, automation tasks, and analyzing data. It's a general-purpose language with widespread applicability instead of a language tailored to a specific domain. Its ease of use and flexibility have contributed to its status as one among the most widely adopted programming languages. Python finds common use in activities like data analysis, visualization, website, and application development. Its user-friendly nature has enabled professionals outside of programming, such as auditors and scientists, to employ it for tasks like financial management due to its approachable learning curve.

1.2.2 CNN

Convolutional neural network (CNN) is class of deep learning patterns specifically designed for image analysis and recognition tasks. It employs multiple layers of specialized neurons, including convolutional and pooling layers, to automatically capture structured features in a hierarchical manner from input images. The convolutional layers apply filters to scan the image, capturing local patterns like edges and textures. Pooling layers downsample feature maps to reduce computation and enhance translation invariance. Dense layers concluding the network process the extracted features to make predictions or classifications. CNNs excel in jobs such as identifying images, object detection, and segmentation due to their ability to learn spatial hierarchies of features, enabling them to recognize complex patterns and structures within images.

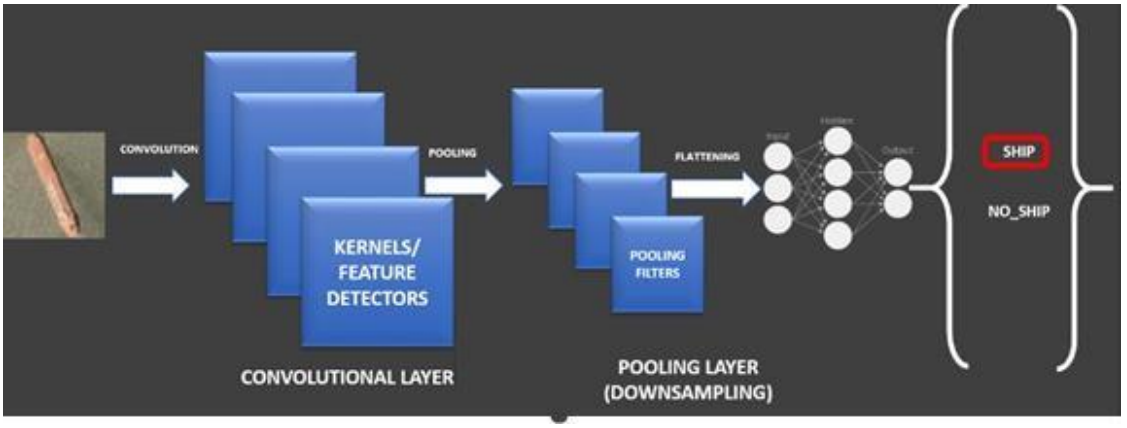

FIGURE 1.1 – CNN model

1.2.3 Machine Learning

Machine learning constitutes a division within the realm of artificial intelligence where algorithms enable systems tolearn from data and improve performance over time without being explicitly programmed.It involves training models on datasets to recognize patterns and make predictions, facilitating tasks like image classification, speech recognition, and data analysis. Machinelearning is essential for the project as it enables the automated and accurate detection of ships within optical satellite images. By Using methods such as Convolutional Neural Network (CNN) to harness their capabilities, the project can train models on labeled datasets to learn complex patterns and features indicative of ships. Subsequently, this model that has undergone training can generalize its understanding to new images, accurately identifying ships amidst various backgrounds and conditions. Without machine learning, manual identification of ships would be time-consuming and error-prone, whereas a well-trained model enhances efficiency and effectiveness, allowing for real-time ship detection and analysis on a larger scale.

1.2.4 Jupyter notebook

The Jupyter Notebook is a web-based interactive application that provides a flexible platform for generating and distributing documents that blend real-time code execution, text with formatting, visual representations, and informative descriptions. This distinctive environment is especially beneficial for professionals engaged in domains like data science, research, and education. It facilitates smooth integration of coding in multiple programming languages like Python, R, and Julia alongside explanatory content, mathematical equations, and interactive graphs. By eliminating the traditional boundaries between code and documentation, Jupyter Notebook supports an iterative process of exploration, analysis, and idea sharing. This enables collaborative creation, experimentation, and presentation of intricate computational workflows and discoveries within a unified interface.

1.2.5 Libraries requires

- **OpenCV:** OpenCV (Open Source Computer Vision Library) is an open-source computer vision and image-processing library. It offers a vast collection of tools and algorithms for tasks like picture and video analysis, feature extraction, object detection, and more. In the project, it's used extensively regarding image processing assignments, like outlining bounding boxes around identified objects ships, converting betweencolor spaces, and enhancing image quality. It significantly contributes to the project's ability to process and analyze images for ship detection

- **Pandas**: Pandas is used for data manipulation and analysis. While not explicitly mentioned in the provided code, Pandas could be used for organizing andhandling tabular data related to the project, such as ground truth ship coordinatesor logging results.
- **NumPy:** NumPy is used for numerical computations. It enables efficient manipulation of arrays and mathematical operations. In the project, it could be used for various numerical operations related to image processing and data analysis.
- **Keras**: Keras provides a high-level neural networks API. Although not seen in the code snippet, Keras could become accustomed to build, train, assess and analyze deep learning models for ship detection, simplifying the process of creating and experimentingwith different neural network architectures**.**
- **TensorFlow:** TensorFlow is used for machine learning tasks. It's utilized to load and utilize a pre-trained CNN model for ship detection, leveraging its efficient computation capabilities on CPUs and GPUs.
- **Pillow**: Pillow is used for image processing tasks. It's employed to open, manipulate, and save image files. In the project, it's used to handle image data and convert between different color spaces**.**
- **Seaborn:** Seaborn is used for data visualization. Although not seen in the code, Seaborn could potentially be used to create aesthetically pleasing and informative visualizations of data related to the project, enhancing the understanding of results and analysis.
- **scikit-learn (sklearn):** Scikit-learn is used for machine learning tasks. It could potentially be used to preprocess data, evaluate model performance, and perform additional analyses related to the project
- **Matplotlib:** Matplotlib is a versatile plotting library. Although not explicitly mentioned, Matplotlib could be used to visualize images, display graphs, or present other visual representations of data, aiding in result visualization and analysis.

1.3 Project Aims and Objectives

Aim:

The aim of this project is to develop a robust deep learning-based detection system techniques and provide a user-friendly web application to demonstrate the ship detection results. The project aims to leverage Convolutional neural network (CNN) Regarding the identification of ships in visual images and enhance the accuracy of ship identification to identify the ships in the image and mark them.

Objectives:

- **Algorithm Design and Development:** Design and implement innovative computer vision algorithms using customized machine learning methods designed for the identification of ships through the analysis of optical satellite pictures. Explore state-of-the-art methods to handle diverse maritime scenarios and challenges in optical images.

- **High Detection Accuracy**: Achieve a high level of detection accuracy by training and fine-tuning the detection model using a comprehensive dataset. Focus on addressing complexities such as occlusions, varying ship orientations, and intricate backgrounds.

- **Quantitative Performance Evaluation:** Conduct thorough performanceevaluation using quantitative metrics such as precision, recall, F1-score. Assess thesystem's reliability and effectiveness across a diverse range of real-world operational scenarios.

- **Developing a front end:** The final objective of the project is to use the trained model to develop a front end application that provide a user interactivity to utilize the model to detect ships

In summary, this project aims to develop an advanced ship detection system that leverages cutting-edge methods of computer vision and machine learning.The system's accurate ship detection capabilities, robustness in challenging conditions, and integration into maritime operations will contribute to strengthening maritime security and domain awareness.

1. LITERATURE SURVEY

2.1 Research papers

Ship detection from optical satellite images based on visual search mechanism

This paper introduces an innovative technique for ship detection, drawing Inspiration from human visual perception. The aim is to enhance the accuracy and efficiency of ship detection in complex maritime environments. The method comprisesthree main stages: local feature extraction, global context analysis, and decision fusion.In the local feature extraction stage, the Scale-Invariant feature transform (SIFT) algorithm is applied to detect ship-like key points. Subsequently, these points are grouped into clusters that hierarchically to form potential ship candidates. By combining local feature extraction, global context analysis, and decision fusion, the paper presents a fresh approach to ship detection, which demonstrates promising results and potential real-world applications optical satellite image. [1]

A visual Search Inspired computational Model for Ship detection in Optical Satellite Images

This research paper presents an innovative computational model designed to identify vessels using optical satellites images, drawing inspiration from human visual searchmechanism. The authors propose a three-stage framework for ship detection. In the first stage, they generate saliency maps using the Itti-Koch model, which captures visual attention mechanisms, highlighting potential regions containing ships. The second stage involves extracting local features employing the scale-invariant feature transform (SIFT) algorithm, which provides distinctive descriptors for ships, facilitating effective localization and differentiation. Lastly, Support vector machine (SVM) classifier is trained to discern ship and non-ship regions based on the extracted SIFT features. The classifier trained using positive and negative samples, resulting in accurate ship detection.Experimental evaluation using a benchmark dataset demonstrates the model efficacy, achieving high ship detection accuracy with a minimal false positive rate, even in

challenging maritime environments with diverse backgrounds and ship sizes.[2]

A Novel Hierarchical Method Of Ship Detection from Space borne optical Image Based on Shape And Texture Features

This paper presents a hierarchical approach for ship detection from optical images using shape and texture feature. This method comprises three key stages: initial extraction of ship candidates, verification of ship candidates, and final ship detection. The first stage, the authors employ morphological operation-based methods to identify potential ship regions based on their distinctive shape characteristics, reducing the search space and enhancing computational efficiency. Moving on to the ship candidate verification stage, set shape, texture feature are extracted to identified ship regions. These features encompass elongation, compactness, rectangular for shape, and Gray-Level Co-occurrence Matrix (GLCM) for texture. Subsequently, the obtained characteristics are employed to train a Support Vector Machine (SVM) classifier capable of distinguishing true ship regions from false positives. In the final ship detection stage, the trained SVM classifier is applied to remaining potential ship regions, classifying them as either ships or non- ships. Subsequently, the detected ship regions undergo refinement using morphological operations and connected component analysis.[3]

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

This paper presents a deep learning-based technique for ship detection and classification for images captured through optical remote sensing. The authors propose a two-stage methodology that integrates ship detection and classification. In the beginning phase, a Convolutional neural network (CNN) is trained to identify ships by analyzing image patches. This CNN is trained using a dataset of annotated images to differentiate between regions containing ships and those without. The ship detection stage successfully pinpoints the areas in the image where ships are present. In the subsequent stage, a separate CNN is trained to categorize the detected ship regions into distinct ship types or classes. The classification model is trained using a dataset of labeled ship images representing various classes. This phase enhances the ship detection results by providing detailed information

about the specific ship types present in the image. Experimental outcomes demonstrate the efficacy of the approach for ship detection and classification. The deep learning-based technique achieves high accuracy in ship detection and proficiently categorizes ships into different classes.[4]

Deep Learning Based automatic detection of Ships: An Experimental Study Using Satellite Images

This paper describes An Experimental Analysis of Satellite Images presents a study conducted for the purpose of experimentation on the automatic finding of ships employing the methodology of deep learning applied to satellite images. The authors aim to create a precise and efficient ship detectionsystem for maritime surveillance and vessel monitoring. The study focuses on the utilizingConvolutional Neural Network (CNN) for ship detection. The authors suggest an integrated deep learning framework which includes pre-processing steps, data augmentation, and CNN- based ship detection models. They assess how well the performance of the models using benchmark datasets to compare the results with traditional ship detection methods. The experimental outcomes demonstrate the deep learning-based approach outperforms traditional methods, achieving high accuracy in ship detection. The study highlights exploring the capabilities of deep learning methods in the automated identification of ships, that providing valuable understanding for the development of advanced ship detection systems using satellite imagery. [5]

Detecting inshore ships in satellite images

This paper introduces a method for detecting inshore ships using satellite images utilizing computer vision techniques. The authors address the challenge of detecting ships near the shore, where they may be partially occluded or exhibit complex visual patterns. The proposed approach involves a multi-stage process. First, a backgroundsubtraction technique is applied to remove the static background and retain only the movingobjects, potentially including ships. Then, a region proposal network is employed to generate candidate ship regions. The regions are further enhanced using shape and textureanalysis to filter out false positives and improve accuracy.

Experimental evaluation on a dataset of satellite images demonstrates the efficiency of the proposed method in detectinginshore ships. This method attains a high level of accuracy in detection. even in challenging scenarios with varying ship sizes, occlusions, and complex backgrounds. The paperhighlights the potential application in surveillance, vessel traffic management, and maritime security. The detection of inshore ships contributes to improved situational awareness and better understanding of maritime activities in coastal areas that can separateships that are onshore. [6]

A Complete Processing Chain for Ship Detection Using optical Satellite Imagery.

This paper provides comprehensive approach for detecting ships in satellite images. The authors strive to create a system that is both effective and precise that can automatically identify and locate ships in large-scale optical satellite data. The processing chain described in paper encompasses several key stages. Initially, image preprocessing is employed to enhance the quality of the satellite imagery, including noise reduction and contrast enhancement. Next, advanced image segmentation method is utilized to separate ship pixels from the background, enabling focused analysis on ship regions. Following segmentation, the paper focuses on feature extraction. Unique geometric, textural, and contextual features are computed from the ship regions, capturing important ship characteristics such as size, shape, and texture. These features serve as a descriptive representation of the ships and contribute to subsequent classification. For classification, authors employ ML algorithm such as support vector machines (SVM) or artificial neural networks (ANN). These algorithms are trained on a labeled dataset, enabling them to learn patterns and discriminate between ship and non-ship samples depending on the extracted attributes. The proposed ship detection feature is evaluated using real-world satellite images, demonstrating its effectiveness. The results indicate high accuracy and efficiency in ship detection, showcasing the potential of the processing chain to various applications in maritime surveillance and remote sensing.[7]

Ship Detection in high resolution Optical Imagery Based on Anomaly Detector and Local Shape Feature

This paper introduces ship detection in high resolution satellites. The authors propose a novel approach that combines an anomaly detector with local shape attributes to achieve accurate and efficient ship detection. The methodology presented in the paper combination of two main components. First, an anomaly detector applied to identify regionsof the image that deviate from the normal background. This detector is trained on a datasetof non-ship regions to learn the characteristics of the background and identify anomalies, which often correspond to ships. The second component focuses on extracting local shape features to the detected anomalies. These features capture the distinctive shape characteristics of ships, such as elongation and compactness. Various shape descriptors, such as circularity or rectangularity, are computed to create a descriptive depiction of the detected anomalies. To classify the detected anomalies as ships or non-ships, a machinelearning algorithm, from a support vector machine (SVM), is employed.

The algorithm is learned on a labeled dataset to learn the patterns and distinguish between ship and non-ship instances depending on the extracted local shape features. The proposed shipidentification method is evaluated using high- resolution optical satellite images, and the results demonstrate its effectiveness in accurately detecting ships while minimizing false detections. The combination of the anomaly identifier and local shape features contributes to improved ship detection performance. In summary, the paper introduces a ship identification method for high-resolution optical images utilizing an anomaly identifier and local shape features. The approach showcases promising results and offers a valuable contribution to the field of remote maritime surveillance and defense. [8]

Accurate Ship Detection Using Electro-Optical Image-Based Satellite on Enhanced Feature and Land Awareness.

This paper proposes an approach for precise ship detection utilizing electro-optical satellite images. The authors aim to improve accuracy in ship detection by enhancing extracted features in the images and incorporating land awareness. The proposed approach involves several steps. Firstly, image-preprocessing techniques are employed to enhance the ship features and suppress the background noise. Then, a feature extraction method is utilized to extract discriminative ship features from the preprocessed images. These attributes are subsequently used to train a model using machine learning algorithms. To further enhance

the ship detection efficiency, the authors introduce a land awareness component. This component incorporates knowledge about land characteristics, such as coastline information, to improvise the ship detection results and reduce false positives. Experimental results provide an effectiveness to the proposed approach in accurately detecting ships in electro-optical satellite images. The approach offers potential applications in maritime surveillance, environmental monitoring, and maritime security. [9]

Inshore ship detection In high resolution Satellite Images: approximation of harbors using sea-land segmentation.

This paper presents a method to detect ship in high resolution satellite pictures specifically in inshore area. The authors suggest an approach that approximates harbors by utilizing sea-land segmentation to enhance ship detection accuracy. The method involves several steps. Firstly, a sea- land using a segmentation algorithm is applied to thesatellite images to distinguish between water and land regions. This segmentation processhelps in identifying potential harbor areas. Next, a batch of geometric and texture-based features are extracted from the segmented regions. These characteristics are subsequently employed to traina machine learning model detect ship. To further improve the accuracy, the authors introduce an approximation technique that refines the detection result by considering spatial distribution and layout characteristics of the identified harbor areas. Experimental results indicate that the proposed method effectively detects ships in inshore regions with high accuracy. The approach has potential applications in maritime surveillance, port management, and coastal security. [10]

Ship Detection In high resolution Optical Imagery Based on Anomaly Detector and Local Shape Feature

This paper published in the journal Remote Sens presents an approach to shipdetection of multispectral satellite images, particularly in complex environmental conditions. The authors address the challenge of accurately detecting ships when various attributes such as cloud cover, sun glint, and sea clutter affect the image quality. The proposed approach involves multiple steps. Initially, image-preprocessing techniques are applied to enhance the ship features and mitigate the impact of environmental conditions.

Then, a spectral and spatial features are exported from the preprocessed image. These properties capture the unique characteristics of ships, enabling differentiation from the background. To further improve ship detection accuracy, a machine learning algorithm is utilized to train a ship classifier using the extracted features. The classifier is capable of distinguishing ships from other object noise in the multispectral satellite images. Experimental outcomes demonstrate efficacy of the suggested method in ship detection under complex environmental properties. This approach has potential applications in maritime surveillance, maritime traffic monitoring, and emergency response operations. [11]

Ship Detection from optical Satellite Images via deep learning

This paper presents a significant contribution to identify utilizing optical satellite images to track ships deep learning techniques. In the context of the authors propose an approach that leveragesconvolutional neural networks (CNNs) to achieve accurate ship detection. They adopt a region proposal network (RPN) to generate candidate ship regions, followed by fine-tuninga Faster R-CNN architecture. To address the challenges of imbalanced ship-background data, they introduce a novel anchor sampling strategy. Their method effectively learns discriminative features to identify ships amidst complex maritime backgrounds. Theproposed approach demonstrates notable performance improvements in terms of precision,recall, and F1 score, confirming the capability of deep learning in identifying ships. tasks in optical satellite imagery. This study's findings contribute to advancing the capabilities to automated ship detection systems, crucial for various maritime applications. [12]

Ship Detection In optical remote Sensing Images: A survey

The paper presents a comprehensive overview on ship detection techniques in the context of remote sensing image. The authors discuss various aspects of ship detection, including challenges posed by diverse environmental conditions, ship types, and image resolutions. They categorize ship detection methods into different approaches, such as traditional handcrafted feature-basedmethods and more recent deep learning-based approaches.

The paper provides insights into feature extraction techniques, dataset availability, and evaluation metrics commonly used in ship detection research. By surveying the advancements in ship detection, this paper serves as a valuable resource for researchers and practitioners interested in understanding the evolution of their applications in the context of optical remote sensing imagery.[13]

Ship Detection In Remote Sensing Images Using convolutional neural network

This paper leverages the power of convolutional neural network (CNN). The authors address the challenge of identifying ships within complex maritime environments by designing a CNN-based architecture. They prepare the remote sensing image for analysis and extract relevant features using a batch of convolutional and pooling layers. The proposed network was trained on a dataset of annotated ship instances and then evaluated on unseenimages. The results illustrate the efficiency of the approach, achieving accurate andefficient ship detection. This research contributes to the growing field of deep learning applied to remote sensing tasks, offering a promising solution for automated ship detection from optical image. [14]

Ship Detection From optical satellite images via deep learning.

The paper presents a approach for ship detection in optical satellite images by deep learning techniques. The authors address the challenge of detecting ships in complexscenes by leveraging convolutional neural network (CNN) and transfer learning from pretrained models, specifically VGG16 and ResNet50 architectures. The proposed approach involves extracting image patches and employing an attention mechanism to highlight ship-related regions for improved localization. Through extensive experiments, the authors demonstrate the effectiveness of their method in accurately detecting ships evenin scenarios with varying ship sizes, orientations, and environmental conditions. The studyshowcases potential of deep learning in advancing ship detection capabilities of optical satellite imagery, contributing to the broader field of remote sensing and object recognition.[15]

2.1. Existing System And Proposed System

2.1.1 Existing system

The current AIS (Automatic Identification System) system operates as a vital mechanism for ship detection, employing a distinctive approach that ensures maritime safety and efficient vessel tracking. In this system, each ship is equipped with an AIS transponder, which transmits a continuous stream of essential information, including the ship's identity, position, course, speed, and other pertinent details.

When operational, the AIS system serves as a dynamic network, with receiving stations strategically positioned along coastlines and aboard satellites. These stations capture and process the transmitted AIS signals, creating an intricate web of real-time vessel data. This amalgamated data stream is subsequently utilized to precisely identify ships, monitor their movement patterns, and maintain situational awareness within maritime domains.
The heart of the AIS system lies in its ability to foster seamless communication between vessels and shore-based or satellite-based receiving stations. Through a series of standardized data protocols and radio frequency transmissions, the AIS-equipped ships continuously share their operational status, facilitating comprehensive maritime domain awareness. This functionality proves especially beneficial for collision avoidance, search and rescue operations, and overall maritime traffic management.

However, it's noteworthy that the AIS system does have its limitations. It heavily relies on the willingness of vessels to transmit accurate information, leaving room for discrepancies or errors in reporting. Additionally, the AIS signals have a limited range, restricting their effectiveness in more remote or densely populated maritime regions and if their is a malfunction in the receiving stations then the ship can not be identified and the transponder can be switched off that can make the ship invisible.

To overcome this drawback in vessel tracking Its reliance on real-time communication and standardized data exchange ensures that vessels' movements are meticulously monitored, contributing to enhanced navigational awareness and streamlined maritime operations using satellite is a viable option to detect ships.

2.1.2 Proposed System

The envisaged system for ship detection through optical satellite images introduces an innovative paradigm that leverages cutting-edge technologies to enhance maritime surveillance and security. This unique approach harnesses the power of optical imagery to identify and track ships, enabling comprehensive coverage of vast maritime regions. The proposed system encompasses several intricate stages, each contributing to the accuracy and its level of efficacy of ship detection.

1. Data Acquisition and Preprocessing

At the core of the system lies the collection of high-resolution optical satellite images from a diverse array of sources. These images are meticulously preprocessed, undergoing noise reduction, radiometric calibration, and geometric rectification to ensure data accuracy. The resulting refined images serve as a foundation for subsequent analysis.

2. Feature Extraction and Pattern Recognition

In this phase, advanced computer vision techniques come into play. The system employs sophisticated feature extraction algorithms to identify distinguishing characteristics of ships. These features may include shape, size, color, and texture patterns that collectively define ship signatures within the optical images.

3. Machine learning Model Integration

To facilitate the ship detection process, a machine-learning model is meticulously trained. This model encompasses neural network structures characterized by profound learning the capability to learn intricate ship features from a diverse dataset. The training involves iterative optimization of the model's parameters, ensuring it generalizes well to new and unseen optical images.

4. Classification and Localization

Once the machine-learning model is trained, it is employed for ship classification and localization. The system identifies ship candidates within the optical images and assigns probabilities to their presence. Localization techniques enable the system to pinpoint the

exact spatial coordinates of detected ships, contributing to precise situational awareness.

5. Post-processing and False Positive Mitigation

To improve the precision of ship detection, a post-processing stage is introduced. This step involves refining the detection results by eliminating false positives through statistical analysis, spatial context, and object tracking. This guarantees that only authentic ship instancesare reported, minimizing potential errors.

6. Performance Evaluation and Validation

The effectiveness of the system is rigorously evaluated through comprehensive performance metrics such as precision, recall, and F1-score. The system's accuracy and reliability are quantitatively assessed across various scenarios and environmental conditions.

In essence, the proposed ship detection system utilizing optical satellite images represents a transformative leap in maritime surveillance capabilities. By amalgamating optical imagery, advanced computer vision, and machine learning, the system empowers authorities to safeguard maritime territories with unprecedented precision and efficiency.

2.2 Feasibility Study

This block includes assessing the feasibility of the undertaking and its practicality presenting a business proposal that outlines a basic project plan and cost estimates. During the viability of the suggested system needs to be assessed evaluated to ensure it won't burden the company. Understanding the core system requirements is crucial in conducting the feasibility study. The feasibility analysis depends upon number three elements:

- ❖ feasibility with regard to Economics
- ❖ feasibility with regard to Technology
- ❖ feasibility with regard to Operations

2.2.1 Feasibility with regard to Economics

The economic feasibility of implementing ship detection utilizing optical satellite images is a critical consideration. While the initial investment in acquiring high resolution optical satellite imagery and developing the advanced detection system may be substantial, the long-term benefits outweigh the costs. The enhanced maritime surveillance capabilities lead to improved safety, efficient traffic management, and reduced response times in critical situations, ultimately resulting in potential savings in terms of disaster mitigation and operational efficiency. Additionally, the system's integration into existing maritime surveillance frameworks reduces the need for extensive infrastructure overhaul, optimizing resource allocation and cost-effectiveness.

2.2.2 Feasibility with regard to Technology

From a technical standpoint, the ship identification system utilizing optical satellite images exhibits promising feasibility. Advances in computer vision, machine learning, and the advancement of deep learning methods has facilitated the creation to highly accurate ship detection algorithms. The integration of these technologies with the optical satellite image provides a powerful means of identifying ships even amidst complex backgrounds and varying lighting conditions. The technical prowess of the system is further supported by its compatibility with existing maritime surveillance infrastructures, ensuring a smooth transition to enhanced ship detection capabilities without major technical hurdles. Rigorous testing and validation protocols ascertain the system's robustness and reliability, affirming its technical viability.

2.2.3 Feasibility with regard to Operations

The operational feasibility of the ship detection system is a key aspect in ensuring its successful adoption. The system's ability to seamlessly integrate into existing maritime surveillance operations and its real-time functionality align well with the operational requirements of naval authorities and maritime agencies. The proposed system's autonomous ship detection capabilities complement traditional methods, freeing Allocate human resources to focus on tasks of greater strategic importance. while maintaining vigilant maritime monitoring. Additionally, the system's capacity to handle large volumes of optical satellite imagery, coupled with robust post-processing mechanisms, enhances operational efficiency and ensures accurate detection outcomes.

2.3 Software and Hardware Specification

Hardware Requirements

The necessary hardware specifications are quite simple, as the computational demands for the ship detection model used are generally not resource-intensive. Here are the recommendedhardware specifications:

- **Processor:** A modern processor with at least a dual-core or quad-core configuration would suffice. Examples include Intel Core i5 or AMD Ryzen 5 processors. The system should have a clock speed of 2.0 GHz or higher to ensure efficient processing.

- **RAM:** A minimum of 4 GB of RAM is recommended for smooth execution of the project. However, if working with larger datasets of images, upgrading to 8 GB or more may enhance performance.

- **Storage:** Sufficient storage space is required to store the image dataset, the trained CNN model, and other project-related files. A minimum of 100 GB of available storage is recommended to accommodate the dataset and related resources.

- **Operating System:** The project can be implemented on various operating systems, such as Windows, macOS, or Linux. Choose an operating system that is compatible with the chosen machine learning framework and programming language.

- **Graphics Processing Unit (GPU):** While the CNN algorithm does not heavily rely on GPU processing, using a GPU can speed up training and inference processes, especially when working with larger datasets. If available, a mid-range GPU, such as NVIDIA GeForce GTX series or AMD Radeon RX series, can give a performance boost.

- **Internet Connectivity:** A stable internet connection is important for dataset collection, updates, and access to additional resources, such as language-specific libraries or pre-trained models.

Software Requirements

The software requirements encompass the necessary tools, libraries, and frameworks to implement and execute the solution. Here are the recommended software requirements:

- **Programming Language:** The project can be implemented using a programming language with machine learning capabilities, such as Python is highly recommended due to its extensive machine learning libraries and ease of use.

- **Integrated Development Environment (IDE):** An IDE provides a comprehensive development environment and facilitates efficient coding and debugging. Popular options for Python include PyCharm, Jupiter Notebook, or Visual Studio Code having python extensions.

- **Dataset:** Collect a representative dataset of Images depicting ships and images lacking them any ships train and evaluate the CNN Model. The dataset can be sourced from public repositories or acquired through data collection techniques.

- **Machine learning libraries:** Machine learning libraries that facilitate the implementation of algorithms, data pre-processing, model training, and evaluation. Commonly used libraries include scikit-learn, keras,tesorflow,PIL

- **Image pre-processing tools:** Image pre-processing Utilized for the purpose of preparing images for machine learning algorithms, tools are employed to clean, normalize, and convert the images into a compatible format. This process might encompass methods like tokenization, resizing the images, removing blurred images.

- **Machine learning algorithms:** Implement and experiment with CNN and Libraries like scikit-learn, keras and tensorflow are used for the implementations ofthese algorithms.

- **Evaluation metrics:** Calculate and Assess the effectiveness of the machine learning models by employing suitable evaluation metrics like accuracy, precision, recall,

F1-score. These metrics can be computed using functions provided by libraries like scikit-learn.

- **Documentation and Visualization:** Utilize tools for generating documentation and visualizations to enhance understanding and present results. Python libraries like Jupyter Notebook, matplotlib, or seaborn can be helpful for generating graphs and visual representations of the project's findings.

- **Dependencies and Package Managers:** Use package managers such as pip (Python) to manage and install required dependencies and libraries.

- **Deployment:** Consider the software requirements for deploying the trained models into a production environment or integrating them with existing APIs. This may involve libraries like Flask for model deployment and serving.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Functional Requirements

The functions that a system must accomplish are specified by its functional requirements. These requirements depend on the kind of software being developed Additionally, the target audience for the final product. These are expectations for the kind of service the network should provide, how it ought to respond to certain inputs, and how it ought to perform in certain circumstances.

The functional prerequisites for this system are as follows:

Table 3.2 Functional Requirements

FR ID	REQUIREMENT	DESCRIPTION
FR01:	User Interface	Provide a user-friendly web-based interface for users to interact with the system.
FR02:	Image Upload	Option to upload image from storage and a button to pass it through the ship detection model.
FR03	Image Preprocessing	The system should preprocess images to enhance their quality, normalize lighting, and adjust contrast.
FR04	Ship Detection	The system must utilize a trained machine educational algorithm for identification ships in the satelliteimages.
FR05	Ship marking	Detected ship positions should be accurately marked on the images.
FR06	Model Accuracy Assessment:	The system should calculate and Present measurements such as accuracy, precision, recall, and F1-score to analyze the model's performance.
FR07	Result Display	Display the Image that has been altered with enclosed regions and Display the overall count of ships detected from the image.
FR08	Compatibility	Ensure the system works with various image resolutions and formats commonly encountered in optical satellite images.

3.3 Non-Functional Requirements

Requirements that not directly related to the system stated function are known as non-functional requirements. These requirements address aspects related to performance, usability and other essential attributes that contribute for the overall effectiveness and user satisfaction of the system.

The following are some of the system's non-functional requirements:

Accuracy: Define the desired accuracy for the ship detection and mark the ships in the image.

Robustness: Address the system's robustness and ability to handle various image types and work efficiently.

Usability: Evaluate the user-friendliness of the ship detection system, considering factors such as ease of integration into existing user interfaces for monitoring system.

Interpretability: Discuss the interpretability of machine learning model used for ship detection. Insights into the models' detection processes, addressing the requirement for transparency and accountability.

User Friendly: The system was created with all of the users in mind. As a result, thetool is basic and straightforward to use. Can have utility for various purposes easily withlittle effort.

Usability: Because it is easy to navigate the system and does so predictably andwith few delays. The system programme responds effectively and swiftly.

.

4. System Design

4.1 system Architecture

The figure 4.1 below illustrates the suggested design pertaining to the system for the ship detection system. This system design involves defining the architectural components, data flow, user interactions, and overall arrangement of the application. The design focuses on ensuring the seamless integration of various modules to achieve accurate spam Ship detection while providing a user-friendly interface.

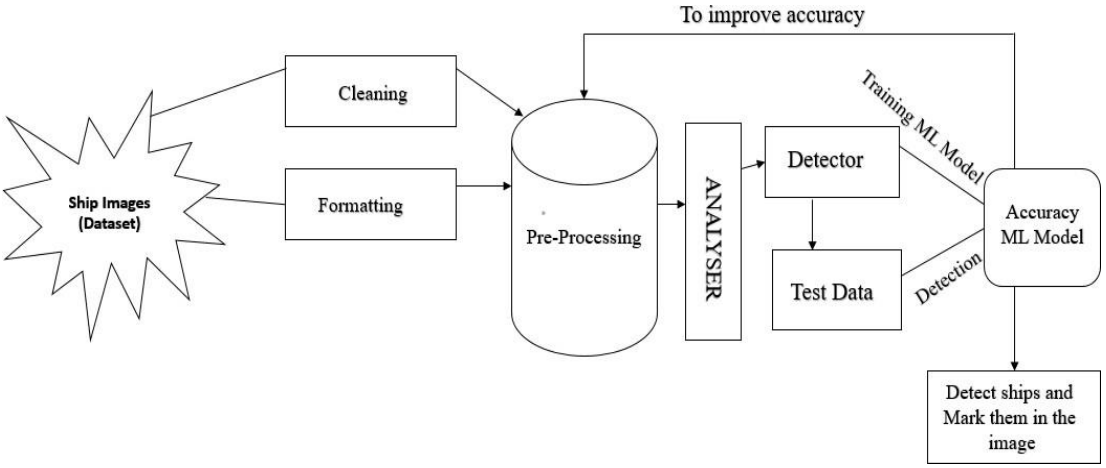

Figure 4.1 System Architecture

4.2 Context Diagram

A context diagram depicts the entire process of developing the model within the system. It showcases how the model is build and flow through various stages, from start to end, during the ship detection process. The initial processes involves collecting the data and preprocessing the data, make it into required format, creating a model, and use it. The diagram visually represents how data is transformed and transmitted between these components, steps involved. It helps in understanding the overall data flow and provides a clear picture of how the system is built to handles and processes images to detect ships.

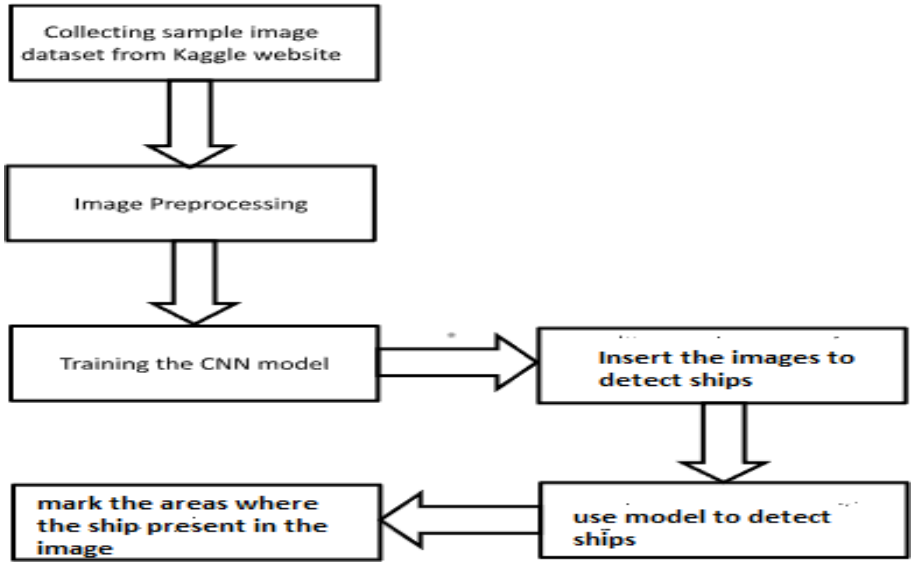

Figure 4.2 Context diagram

5 DETAIL DESIGN

5.1 Use case diagram.

A use case diagram displays the various interactions between system users (actors) and the system itself. It identifies the different functionalities and actions that can be performed within the system. Actors in the diagram may include authorities as users, administrators, or external systems. The use cases represent specific actions or behaviors, such as uploading the image training the machine learning model, or managing the system's settings. The diagram presents a summary of the system's capabilities and helps authorities comprehend the manner in which users engage with. the system to achieve their goals related to ship detection in images messages.

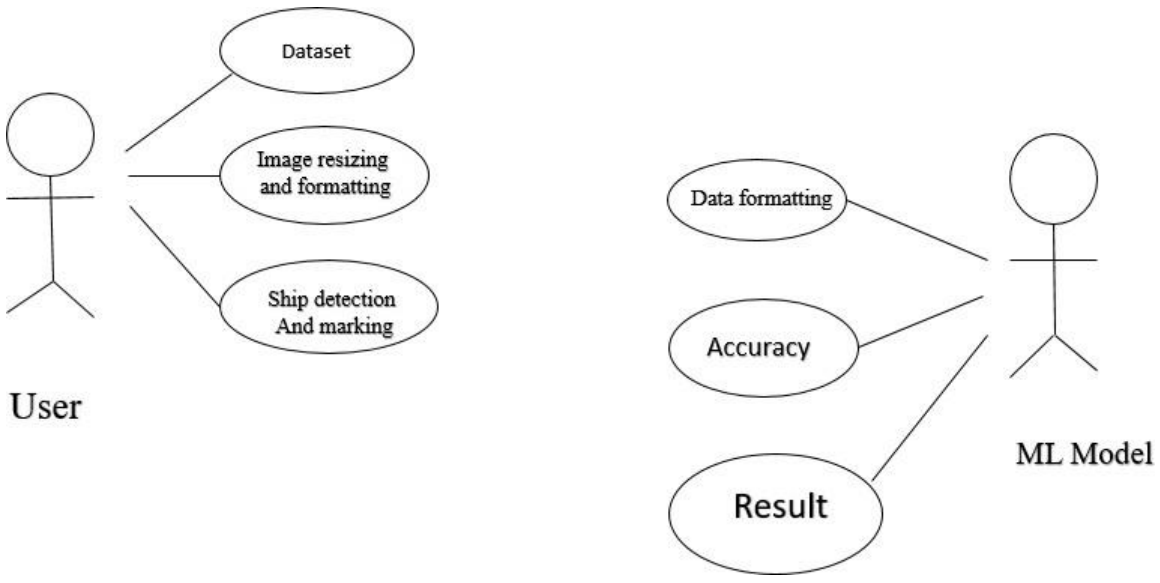

Figure 5.1 use case diagram

5.2 Activity Diagram

An activity diagram depicts the sequential flow of activities and actions within the system. It provides a visual representation of the steps involved in the ship detection process, including data preliminary data preparation, deriving features, model training, and ship identification. The diagram displays the decision points, parallel activities, and loops, highlighting the logical flow of operations. It helps in understanding the overall workflow and the arrangement of activities performed in detecting ships from the image using deep learningtechniques. The activity diagram aids in identifying ships, optimizing the process, and ensuring efficient ship detection from the image.

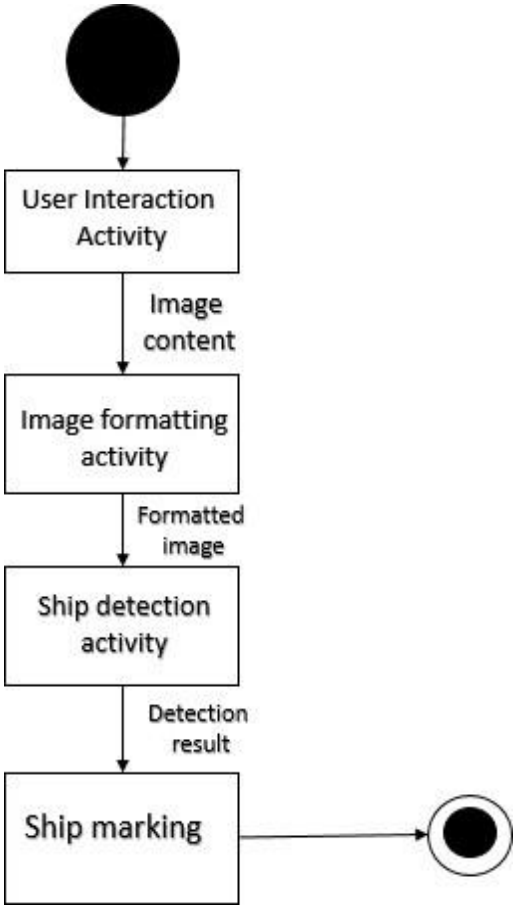

Figure 5.2 Activity Diagram

5.3 Sequence Diagram

A sequence diagram illustrates the dynamic interactions and data exchanges between objects over a specific scenario or use case. It displays the arrangements of events and the order in which image is processed between different components or classes within the system. The diagram typically represents the flow of actions and communications during the ship detection process, from receiving an image to identify ships in it and mark them. It helps to visualize the chronological order of operations, method invocations, and data exchanges, providing a clear understanding on how the system components collaborate to achieve the desired outcome of ship detection in the image uploaded.

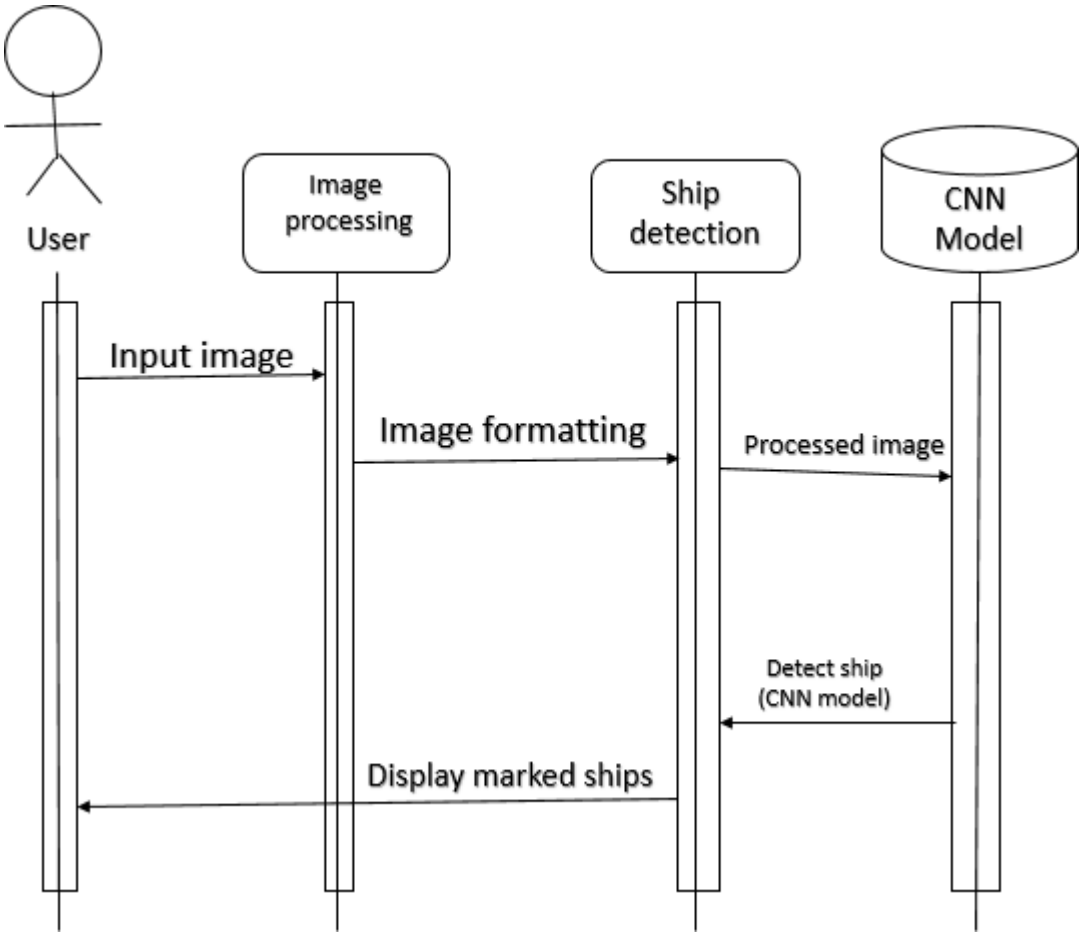

Figure 5.3 Sequence Diagram

6. IMPLEMENTATION

6.1 Snippet Code:

Initially the dataset of ships should be collected and preprocessed by resizing the images, removing the noise for training the model. Next, label the images and make it into a same format image. Next train the model using the images and save the model, later install flask and use the saved model to develop a front end application.

6.1.1 Importing libraries

The code snippet involves importing required libraries for developing a ship detection model these libraries is required for setting up a CNN architecture with Convolutional and Pooling layers, creating a sequential model, and configuring training parameters and for evaluating the effectiveness of the model using confusion matrices and visualization tools like seaborn and matplotlib.

```
Import json, sys, random
Import numpy as np
import tensorflow as tf
Import pandas as pd
from keras.models import Sequential
from keras.layers import Dense, Flatten, Activation, Dropout
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.utils import np_utils
from keras.optimizers import SGD
import keras.callbacks
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, plot_confusion_matrix
import seaborn as sns
Import Matplotlib.pyplot as Plt
from PIL import Image, ImageDraw
```

6.1.2 Data collection and preprocessing

The provided code preprocesses a dataset named 'shipsnet.json' It loads the dataset containing image data and corresponding labels for ship and non-ship classes. It reshapes the image data to match the expected input format for the CNN model. The labels are transformed into one-hot encoded format. Images belonging to ship and non-ship classes are separated for analysis. Finally, the image data is normalized by dividing it by 255 to scale pixel values between 0 and 1, preparing it for training the CNN model for ship detection.

```
f = open('shipsnet.json')
dataset = json.load(f)
f.close()
X = np.array(dataset['data']).astype('uint8')
y = np.array(dataset['labels']).astype('uint8')
pd.Series(y).value_counts()
X_reshaped = X.reshape([-1, 3, 80, 80])

X_reshaped.shape

X_reshaped = X.reshape([-1, 3, 80, 80]).transpose([0,2,3,1])
X_reshaped.shape

y_reshaped = tf.keras.utils.to_categorical(y, num_classes=2)
y_reshaped.shape

imgs_0 = X_reshaped[y==0]
imgs_1 = X_reshaped[y==1]

X_reshaped = X_reshaped / 255
```

6.1.3 Splitting The Dataset

The code splits the preprocessed ship detection dataset into training and test sets. It first divides the data into training (80%) and test (20%) sets. The `random_state` parameter ensures reproducibility in the split.

```
X_train_full, X_test, y_train_full, y_test = train_test_split(X_reshaped,
y_reshaped, test_size=0.20, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train_full, y_train_full,
test_size=0.25, random_state=42)
```

6.1.4 Network design for CNN

The provided code defines a structure for Convolutional Neural Network (CNN) for ship detection. It consists of multiple convolutional and pooling layers, followed by dropout regularization, leading to fully connected layers. The last layer using softmax activation 2 units corresponds to ship (class 1) or background (class 0) prediction. This architecture processes input images of size 80x80 pixels, progressively extracting features and reducing spatial dimensions to identify ships within images.

```
model = keras.Sequential()

model.add(Conv2D(32, (3, 3), padding='same', input_shape=(80, 80, 3),
activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) #40x40
model.add(Dropout(0.25))

model.add(Conv2D(32, (3, 3), padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) #20x20
model.add(Dropout(0.25))

model.add(Conv2D(32, (3, 3), padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) #10x10
model.add(Dropout(0.25))

model.add(Conv2D(32, (10, 10), padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2))) #5x5
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(2, activation='softmax'))
```

6.1.5 Training The Model

The provided code trains a ship Stochastic gradient descent employed in a detection model optimizer. It compiles the model with categorical cross-entropy loss and accuracy metric. The model trained for 20 epochs using training data `(X_train, y_train)` and validated on `(X_val, y_val)`. This helps the ship detection model learn to classify ship-related features in images while adjusting its parameters through backpropagation.

```
sgd = SGD(learning_rate=0.01, momentum=0.9, nesterov=True)
model.compile(
   loss='categorical_crossentropy',
   optimizer=sgd,
   metrics=['accuracy'])
history = model.fit(
   X_train,
   y_train,
   epochs=20,
   validation_data=(X_val, y_val),
   verbose=2)
```

6.1.6 Python Flask Implementation

This code defines a Flask web application for a ship detection project. It loads a pre-trained CNN model to detect ships in uploaded images. Users can upload images through the web interface, and the application processes these images using the CNN model to detect ships. Detected ship coordinates are then filtered using Non-Maximum Suppression (NMS), and Bounding boxes are outlined around ships in the processed image. The application displays the original image, processed image with bounding boxes, and detected ship coordinates on the web interface.

```
app = Flask(__name__)
model = load_model('static/models/CNN_TrainedModel.h5')
app.config['UPLOAD_FOLDER'] = 'static/uploads'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

def allowed_file(filename):
   return '.' in filename and filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

@app.route('/', methods=['GET', 'POST'])
def index():
   return render_template('index.html')

@app.route('/about', methods=['GET', 'POST'])
def about():
   return render_template('about.html')

@app.route('/predict', methods=['GET', 'POST'])
def predict():
   step_size = 18 # Default step size
   image_path = None
   ship_coordinates = []

   if request.method == 'POST':
      #step_size = int(request.form.get('step_size'))

      if 'image' not in request.files:
         return render_template('predict.html', step_size=step_size, error='No file
part')

      file = request.files['image']

      if file.filename == '':
```

```
        return render_template('predict.html', step_size=step_size, error='No
selected file')

    if file and allowed_file(file.filename):
        image_path = os.path.join(app.config['UPLOAD_FOLDER'],
'uploaded_image.png')
        file.save(image_path)


        image  =  Image.open(image_path)
        image_rgb = np.array(image.convert("RGB"))

        # Define patch and step sizes
        patch_size = 80


        ship_candidates = []
        for y in range(0, image_rgb.shape[0] - patch_size + 1, step_size):
            for x in range(0, image_rgb.shape[1] - patch_size + 1, step_size):
                patch = image_rgb[y:y+patch_size, x:x+patch_size]
                input_data = patch[np.newaxis, ...]
                prediction = model.predict(input_data)
                if prediction[0][1] > 0.90:
                    ship_candidates.append((x, y))

        # Apply Non-Maximum Suppression
        def non_max_suppression(boxes, overlap_threshold=0.2):
            if len(boxes) == 0:
                return []

            boxes = np.array(boxes)
            x1 = boxes[:, 0]
            y1 = boxes[:, 1]
            x2 = x1 + patch_size
            y2 = y1 + patch_size

            areas = (x2 - x1 + 1) * (y2 - y1 + 1)
            scores = [1.0] * len(boxes)

            sorted_indices = np.argsort(scores)
            selected_indices = []

            while len(sorted_indices) > 0:
                last = len(sorted_indices) - 1
                i = sorted_indices[last]
                selected_indices.append(i)

                xx1 = np.maximum(x1[i], x1[sorted_indices[:last]])
```

```
            yy1 = np.maximum(y1[i], y1[sorted_indices[:last]])
            xx2 = np.minimum(x2[i], x2[sorted_indices[:last]])
            yy2 = np.minimum(y2[i], y2[sorted_indices[:last]])

            width = np.maximum(0, xx2 - xx1 + 1)
            height = np.maximum(0, yy2 - yy1 + 1)
            overlap = (width * height) / areas[sorted_indices[:last]]

            sorted_indices = np.delete(sorted_indices, np.concatenate(([last],
np.where(overlap > overlap_threshold)[0])))

        return [ship_candidates[i] for i in selected_indices]

    # Apply NMS to get final ship coordinates
    final_ship_coordinates = non_max_suppression(ship_candidates)

    # Draw bounding boxes around detected ships
    for coord in final_ship_coordinates:
        x, y = coord
        cv2.rectangle(image_rgb, (x, y), (x + patch_size, y + patch_size), (0, 255,
0), 2)


    processed_image_path = os.path.join(app.config['UPLOAD_FOLDER'],
'processed_image.png')
    cv2.imwrite(processed_image_path, cv2.cvtColor(image_rgb,
cv2.COLOR_RGB2BGR))

    return render_template('predict.html', step_size=step_size,
image_path=image_path, ship_coordinates=final_ship_coordinates,
processed_image_path=processed_image_path)

  return render_template('predict.html', step_size=step_size,
image_path=image_path, ship_coordinates=ship_coordinates)

if __name__== '_main_':
  app.run(debug=True)
```

6.2 Results

The below image explains about, evaluation measures such as precision and recall, F1 score, and support play vital roles in evaluating the model's effectiveness. Precision measures the accuracy of detected ships among all predicted ships, while recall assesses the model's ability to identify all actual ships. The F1 score combines precision and recall, offering a balanced performance measure. Support indicates the number of instances of each class, aiding in understanding data distribution and potential biases in ship detection results.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	591
1	0.97	0.97	0.97	209
micro avg	0.98	0.98	0.98	800
macro avg	0.98	0.98	0.98	800
weighted avg	0.98	0.98	0.98	800
samples avg	0.98	0.98	0.98	800

Figure 6.1 Performance Metrics

The below image explains about Convolutional Neural Network (CNN) and sequential model. It consists of layers like max pooling to reduce spatial dimensions, dropout to prevent overfitting by randomly deactivating neurons, flatten to transform 2D feature maps into a 1D vector, and dense layers for classification. Max pooling reduces data size while retaining important features, dropout improves generalization, flatten prepares data for fully connected layers, and dense layers make predictions based on learned features. This architecture helps the model learn hierarchical patterns in images for accurate ship detection.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 80, 80, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 40, 40, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 40, 40, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 40, 40, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 20, 20, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 20, 20, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 20, 20, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 10, 10, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 32)        102432
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 32)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 5, 5, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 800)               0
_________________________________________________________________
dense (Dense)                (None, 512)               410112
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 1026
=================================================================
Total params: 532,962
Trainable params: 532,962
Non-trainable params: 0
_________________________________________________________________
```

Figure 6.2-Trained Model Results

The below image shows the model graph in a ship detection project that visually represents the performance of the trained ship detection model over different epochs or training iterations. It displays how accurately the model is able to identify ships in images as training progresses. The y-axis indicates the accuracy percentage, while the x-axis represents the number of training epochs. A rising curve on the graph signifies improving accuracy, showing the convergence of the model's learning process.

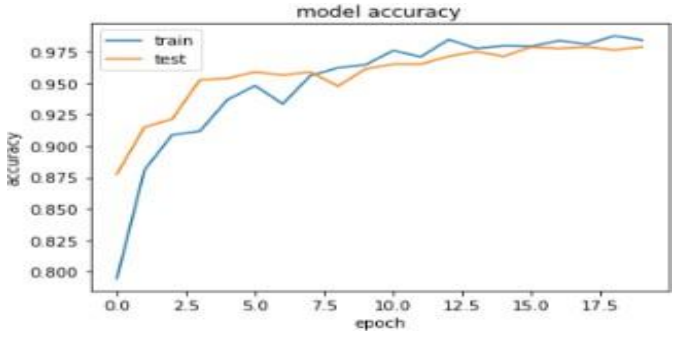

Figure 6.3 Model Accuracy

A confusion matrix in a ship detection project is a tabular representation that summarizes the performance of a machine learning model's predictions. It shows the counts of true positive, true negative, false positive, and false negative predictions for ship and non-ship classifications. This matrix helps to analyze the model's accuracy, precision, recall, and F1-score, providing insights into its effectiveness in correctly identifying ships and background elements in images.

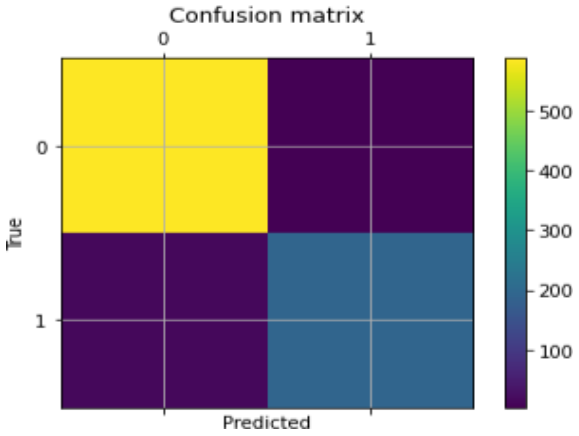

Figure 6.4 Confusion Matrices of the model

The below figure is the front end implementation of the ship detection system, where we can give input of image to detect ships in it

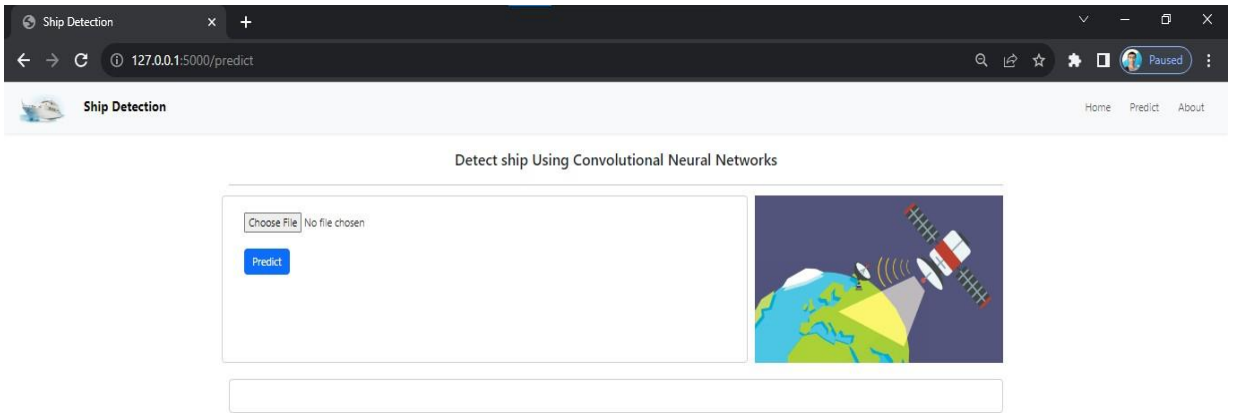

Figure 6.5 Image Upload Module

The below figure is the output of the ship detection system containing the original image and the processed image where the ships are marked along with the location of it

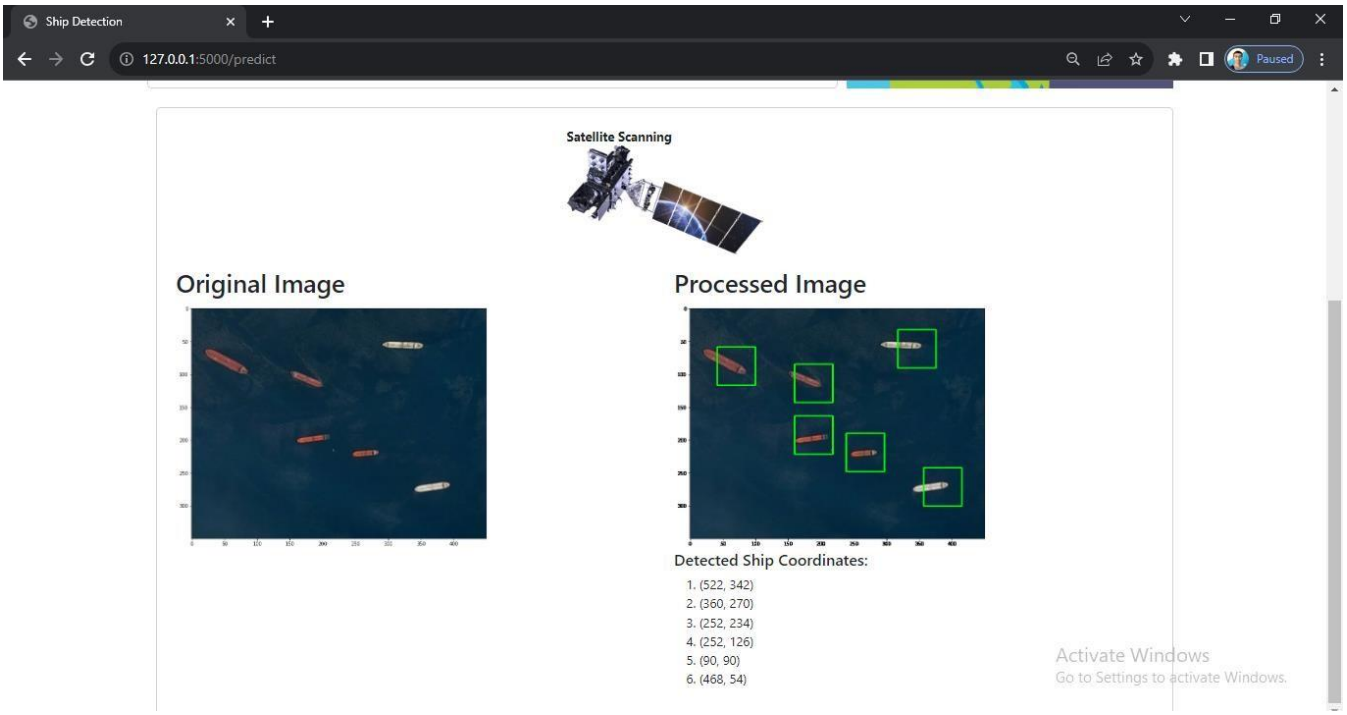

Figure 6.6 result of ship detection

7. SOFTWARE TESTING

Software testing aims to identify errors by examining a piece of work, involving the search for weaknesses or flaws. This process allows for the purpose of evaluating the cohesive functionality of individual parts, as well as the interactions between them, ultimately ensuring that a final product meets user expectations and operates as intended, without any undesirable glitches.

7.1 Types of Testing

- **Functionality testing**

  Functionality testing involves examining the application to ensure it fulfills the specified requirements. Testers also assess the application's workflow during this process.

- **Integration testing**

  This refers to a testing approach where each module is tested separately for bugs and errors before being integrated. The entire application is then tested comprehensively to find any errors that might occur throughout its execution.

- **Unit testing**

  Unit testing entails the generation of test cases to check the accurate functioning of the software's fundamental logic and the generation of valid outputs from given inputs. It necessitates examining the Internal code execution sequence all decision branches within the software. This method includes scrutinizing each individual module of the software, which takes place after the completion of each component and before integration. These specific test procedures heavily rely on existing structural knowledge. In the element level, unit testing assesses distinct business procedures, programs, or system configurations. Its purpose to confirm that each stage of organizational processes produces well-defined output signals that align with the specified requirements.

- **Black Box Testing**

  Black box testing is the practice of evaluating software without examining its code, known as "black box testing". It involves scrutinizing the component under examination without delving into its inner workings such as code, structure, or languages. This technique encompasses diverse facets of the component's performance. Like other testing methods, black box testing requires a dedicated

reference document, like a specification or standards file. This testing approach treats the subject model as an opaque entity, much like a sealed box, disregarding its internal mechanisms. In this manner, the test evaluates how the model responds to inputs and generates outputs, without concern for its internal functioning.

- **Acceptance Testing**

The incorporation of testing into any project's user adoption necessitates engagement from ordinary users. This involvement verifies the alignment of the system with the specified requirements.

7.2 Test cases

Table 7.1 unit testing for launching module

Test case Id	T001
Test case Type	Unit testing
Description	Web application should be able to run when given the link in the browser.
Expected Result	Application launched
Actual result	Application launched
Test case status	Pass

Table 7.2 integration testing for home page

Test Case Id	T002
Test case Type	Integration testing
Description	Once the link generated by the server is put in the browser the home page should be displayed
Expected Result	Home page displayed
Actual result	Home page displayed
Test case status	Pass

Table 7.3 Functionality testing for file window

Test Case Id	T003
Test case Type	Functionality testing
Description	Once we click on the choose file option a window should be displayed having image files
Expected Result	Window with image file displayed
Actual result	Window with image file displayed
Test case status	Pass

Table 7.4 unit testing for choose file module

Test Case Id	T004
Test case Type	Unit testing
Description	Once we click on the choose file option a window should be displayed having image files and the file we choose should be uploaded
Expected Result	Image uploaded
Actual result	Image uploaded
Test case status	Pass

Table 7.5 unit testing for output

Test Case Id	T005
Test case Type	Unit testing
Description	The model should check for the ships detect them and display the marked images with ship location
Expected Result	Ship detected and marked
Actual result	Ship detected and marked
Test case status	Pass

8. CONCLUSION

In this project, we have utilized the deep learning methods and in conclusion, of the ship detection system has successfully demonstrated the potential of combining image processing and object detection for ship detection. The system utilizes a trained convolutional neural network (CNN) model to accurately recognize and detect ships and mark them in the image. By analyzing the shape, size of the ship the CNN model can detect the ship.

The system shows promising results and has practical applications in enhancing maritime surveillance in identifying ships that has been invisible to AIS with the potential to shape the future of surveillance in domain of defense. However, further research and development are needed to refine the system, address any limitations, and explore its wider applications in different contexts.

.

9. FUTURE ENHANCEMENT

Future research directions will prioritize integrating location features into the object detection/classification system to precisely ascertain the ship's position and integrate it with AIS (Automatic Identification System). Furthermore, additional experiments should be conducted to evaluate various sensors, like SARs (Synthetic Aperture Radars), under challenging conditions where visible spectrum imagery is unavailable, such as during nighttime, cloudy weather or foggy conditions. Moreover, exploring multi-modal scenarios by integrating different sensors and leveraging saliency estimation methods not only for ship classification but also for determining the precise positions of identified ships and other objects like yachts, boats, and aircraft will be a key focus. The incorporation and utilization of available HPC (High-Performance Computing) resources will also be a significant aspect of future work.

10. REFERENCES

[1] Yang, Q. Xu, f.gao , L. Hu, "Ship detection from optical satellite images based on visual search mechanism," 2015 ieee international geoscience and remote sensing Symposium(IGARSS), Milan, Italy, 2015, pp. 3679-3682, doi: 10.1109/IGARSS.2015.7326621.

[2]F. Bi, B. Zhu, L. Gao and M. Bian, "A Visual Search Inspired Computational Model for Ship Detection in Optical Satellite Images," IEEE geoscience and remote sensing letters, vol. 9, no.4, pp. 749-753, July 2012, doi: 10.1109/LGRS.2011.2180695.

[3]C. zhu, H. zhou, R. wang and j. guo, "A Novel Hierarchical Method of Ship Detection from Spaceborne Optical Image Based on Shape and Texture Features," in IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 9, pp. 3446-3456, Sept. 2010, doi: 10.1109/TGRS.2010.2046330.

[4] Ying Liu, Hong-Yuan Cui, Zheng Kuang and Guo-Qing Li"Ship Detection and Classification on Optical Remote Sensing Images Using Deep Learning" ITM Web Conf. **Volume** 12, 2017 The 4th Annual international conference on information Technology and Applications (ITA 2017)

[5] **Krishna Patel**, **Chintan Bhatt**,* and **Pier Luigi Mazzeo** "Deep Learning-Based Automatic Detection of Ships: An Experimental Study Using Satellite Images" Received: 28 April 2022 / Revised: 6 June 2022 / Accepted: 23 June 2022 / Published: 28 June 2022

[6]A. H. Özcan and C. Ünsalan, "Detecting inshore ships in satellite images," 2015 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, Turkey, 2015, pp. 1220-1223, doi: 10.1109/SIU.2015.7130057.

[7]Corbane, Christina & Najman, Laurent & Pecoul, Emilien. (2010). A complete processing chain for ship detection using optical satellite imagery. International Journal of Remote Sensing - INT J REMOTE SENS. 31.10.1080/01431161.2010.512310.

[8]Z. Shi, X. Yu, Z. Jiang and B. Li, "Ship Detection in High-Resolution Optical Imagery Based on Anomaly Detector and Local Shape Feature," in IEEE Transactions on Geoscience

and Remote Sensing, vol. 52, no. 8, pp. 4511-4523, Aug. 2014, doi: 10.1109/TGRS.2013.2282355.

[9] Lee, S.-H.; Park, H.-G.; Kwon, K.-H.; Kim, B.-H.; Kim, M.Y.; Jeong, S.-H. Accurate ship detection using electro optical image based satellite on enhanced feature and land awareness. Sensors **2022**, 22, 9491. https://doi.org/10.3390/s22239491

[10] Beşbınar, Beril & Alatan, A (2015). Inshore ship detection in high-resolution satellite images: approximation of harbors using sea-land segmentation. 96432D. 10.1117/12.2194928.

[11] Xie, Xiaoyang & Li, Bo & Wei, Xingxing. (2020). ship detection In Multispectral satellite images under complex environment. remote sensing. 12.792.10.3390/rs12050792.

[12] Chen, K., Wang, X., Jia, J., & Liao, S. (2018). ship detection From optical satellite images via deep learning. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 73-88). DOI: 10.1007/978-3-030-01249-6_5

[13] Zhao, R., Wang, B., & Liu, J. (2019). ship Detection In optical Remote sensing images: A survey. IEEE Geoscience and Remote Sensing Magazine, 7(2), 8-26. DOI: 10.1109/MGRS.2019.2894052

[14] Li, Z., & Zhang, L. (2019). Ship Detection In remote Sensing image using convolutional neural networks. Remote Sensing, 11(20), 2406. DOI: 10.3390/rs11202406

[15] Chen, K., Wang, X., Jia, J., & Liao, S. (2018). Ship Detection From optical satellite Images via deep learning. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 73-88). DOI: 10.1007/978-3-030-01249-6_5

Appendix c: User Manual

Step 1: Open cmd from the folder.

Step 2: Type “pip install –r requirements.txt” and click enter.

Step 3: After installation type app.py.

Step 4: The app.py will give an port number and location.

Step 5: Copy and paste the location in the browser.

Step 6: Click on detect page.

Step 7: Select the image file to detect the ships and click on detect button.

Step 8: The system will show the ships from the image and mark them with location.