RV Educational Institutions ®
RV College of Engineering ®
Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi
Approved by AICTE, New Delhi

Go, change the world

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Synthetic underwater image data set generation using deep neural networks

PROJECT REPORT

Submitted by

Mohamed Moin Irfan	**1RV19CS089**
Prerana Shekar M S	**1RV18CS121**
Purnodeep Rajankar	**1RV19CS122**
Sathvik Gowda M	**1RV18CS146**

Under the guidance of
Prof Manonmani S
Assistant Professor
Dept of CSE
RV College of Engineering

In partial fulfilment for the award of degree of
Bachelor of Engineering
in
Computer Science and Engineering
2022-2023

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the major project work titled ‘***Synthetic underwater image data set generation using deep neural networks*** ’ is carried out by **Mohamed Moin Irfan (1RV19CS089), Prerana Shekar M S (1RV18CS121),Purnodeep Rajankar(1RV19CS122) and Sathvik Gowda M (1RV18CS146)** who are bonafide students of RV Collegeof Engineering, Bengaluru, in partial fulfilment for the award of degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2022-2023. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the major project report depositedin the departmental library. The major project report has been approved as it satisfies the academicrequirements in respect of major project work prescribed by the institution for the said degree.

Signature of Guide	Signature of Head of the Department	Signature of Principal
Prof. Manonmani S	**Dr. Ramakanth Kumar P**	**Dr.K.N.Subramanya**

External Viva

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

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Mohamed Moin Irfan, Prerana Shekar M, **Purnodeep Ranjankar and Sathvik Gowda M** students of eighth semester B.E., department of CSE, RV College of Engineering, Bengaluru, hereby declare that the major project titled ***'Synthetic underwater image data set generation using deep neural networks'*** has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering** during the year 2022-23.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru
Date:

Name **Signature**

1. **Mohamed Moin Irfan(1RV19CS089)**
2. **Prerana Shekar M S (1RV18CS121)**
3. **Purnodeep Rajankar(1RV19CS120)**
4. **Sathvik Gowda M (1RV18CS146)**

ACKNOWLEDGEMENT

We are indebted to our guide**, Prof. Manonmani S,** Assistant Professor**, Dept of CSE** for her wholehearted support, suggestions and invaluable advice throughout our project work and also helped in the preparation of this thesis.

We also express our gratitude to our panel members **Dr. Rajashree Shettar,** Professor & Dean (PG-Circuit Programs) Department of Computer Science and Engineering and **Prof Srividya M S,** Assistant Professor Department of Computer Science and Engineering for their valuable comments and suggestions.

Our sincere thanks to **Dr. Ramakanth Kumar P.**, Professor and Head, Department of Computer Science and Engineering, RVCE for his support and encouragement.

We express sincere gratitude to our beloved Principal,
Dr. K. N. Subramanya for his appreciation towards this project work.

We thank all the **teaching staff and technical staff** of the Computer Science and Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our **family** members and **friends** who provided all the backup support throughout the project work.

TABLE OF CONTENTS

Chapter 3

Software Requirements Specification

Chapter 6

Implementation of Design of Detection and Extraction of Naval Mine Features System

LIST OF TABLES

ABSTRACT

This project focuses on the creation and evaluation of an synthetic dataset for underwater mine detection using Convolutional Neural Networks (CNN) as the machine learning model and Contrast Limited Adaptive Histogram Equalization (CLAHE) for dataset preprocessing. The scarcity and high cost of collecting large-scale underwater mine images make it challenging to develop robust machine learning models. To address this issue, we propose an augmentation-based approach to expand the dataset and enhance model performance.Starting with a limited set of 14 original images of underwater mines, we employ various augmentation techniques to generate a dataset comprising 2100 images. These techniques include rotations, translations, flips, and color transformations. The augmented dataset aims to capture a diverse range of mine variations and environmental conditions.Prior to training the CNN model, we preprocess the augmented dataset using CLAHE and Adam optimizer. This technique enhances the image contrast and improves the visibility of subtle details, enabling the model to better discriminate between mine and non-mine regions.

To evaluate the effectiveness of our augmented dataset and preprocessing technique, we compare the performance of the CNN model trained on our dataset with that of a benchmarked dataset. The benchmarked dataset consists of high-quality images collected from reputable sources. Both datasets are loaded onto the CNN model, and their respective results are compared.The evaluation metrics used for comparison include accuracy, trend of loss, validation loss, MSE and F1-score. By analyzing and comparing the performance of the two datasets, we aim to assess the correctness and suitability of our synthetic dataset and CLAHE preprocessing for underwater mine detection.This project contributes to the field of underwater mine detection by providing a larger and more diverse dataset through augmentation techniques and utilizing CNN with CLAHE preprocessing. The findings of this study will aid in the development of more accurate and robust machine learning models for effective underwater mine detection and countermeasures.

ACRONYMS

- ML - Machine Learning
- CLAHE -Contrast Limited Adaptive Histogram Equalization
- RVUMR-14 – Rashtreeya Vidyalaya Underwater Mine Research 14
- CIFAR-10 -Canadian Institute for Advanced Research -10
- MSE - Mean Squared Error
- DL - Deep Learning
- CNN -Convolutional Neural Network
- SRS - Software Requirement Specification
- DFD - Data Flow Diagrams
- SDLC - Software Development Life Cycle

CHAPTER 1-
AN INTRODUCTION TO DETECTION AND EXTRACTION OF NAVAL MINES

CHAPTER 1
Introduction

Underwater mine detection is a critical area of research in maritime security. The development of accurate and robust detection models is hindered by the scarcity of large-scale underwater mine datasets and the lack of diversity in existing datasets. This project aims to address these limitations by creating an augmented dataset for underwater mine detection using augmentation techniques.The motivation behind this project stems from the need to enhance underwater mine detection capabilities. By utilizing machine learning models, such as CNN, and preprocessing techniques like CLAHE, we aim to improve the performance of these models in detecting underwater mines.The primary objective of this project is to create an augmented dataset comprising 2100 images from 14 original images using augmentation techniques. We employ rotations, translations, flips, and color transformations to generate a diverse range of mine variations and environmental conditions. Furthermore, we preprocess the dataset using CLAHE to enhance image quality.

To validate the effectiveness of the augmented dataset, we compare the performance of a CNN model trained on our dataset with that of a benchmarked dataset. This comparison is based on evaluation metrics such as accuracy, precision, recall, and F1-score.The project's scope includes specific types of underwater mines, selected augmentation techniques, and the evaluation metrics used. The methodology involves training and evaluating the CNN model on both the augmented dataset and the benchmarked dataset.Overall, this project aims to contribute to the field of underwater mine detection by providing an expanded and diverse dataset using augmentation techniques. By evaluating and comparing the performance of the datasets, we seek to enhance the accuracy and reliability of machine learning models for effective underwater mine detection and countermeasures.

1.1 State of Art Developments

Underwater mine detection is a critical area of research due to its implications for maritime security. This section provides an overview of the current state of art developments in underwater mine detection techniques, including the utilization of machine learning and deep learning models as follows:

- CNN is used to build the model.
- The CIFAR-10 dataset is used to validate.
- The masks are used as ground truth in semantic segmentation.
- Various precision metrics like MSE, F1_SCORE,and accuracy are used to evaluate the model.

1.2 Problem Statement

The scarcity of large-scale underwater mine datasets poses a significant challenge in developing accurate and robust detection models. Additionally, the lack of diverse underwater mine images further hinders the performance of existing models.Hence, to create a Synthetic dataset for different types of mines and train the model for accurate prediction of the mines.

1.3 Motivation

The motivation behind this project stems from the necessity to improve underwater mine detection capabilities through the development of reliable machine learning models. The current datasets does not include the different types of mines but rather relay on the source being a non-movable metal object or moving sea animal. Hence, the idea of training the model for the sole purpose of underwater mine detection with different classes of mines with country origins this research aims to enhance the performance of underwater mine detection systems.

1.4 Objectives

The primary objective of this project is to create an dataset for underwater mine detection using augmentation techniques and perform annotation to create masks for the dataset. Additionally, the project aims to validate the effectiveness of the augmented dataset by comparing the performance of a CNN model trained on the augmented dataset with a benchmarked dataset.

1.5 Scope

Collection of such classified information is itself challenging. To collect various types of underwater mines and country of origin. Further, creating dataset from 14 classes of mine with each class having 150 images using different augmentation techniques like fliping, rotation, grayscale and transform. To create masks for all the images by annotating them. Further, to evaluate the dataset by training the CNN model with evaluation metrics like training accuracy, validation accuracy, validation loss, trend of accuracy and trend of loss. Also, to predict a image and evaluate the correctness, the metrics like MSE, F1_score, precision, recall and confusion matrix are used.

1.6 Methodology

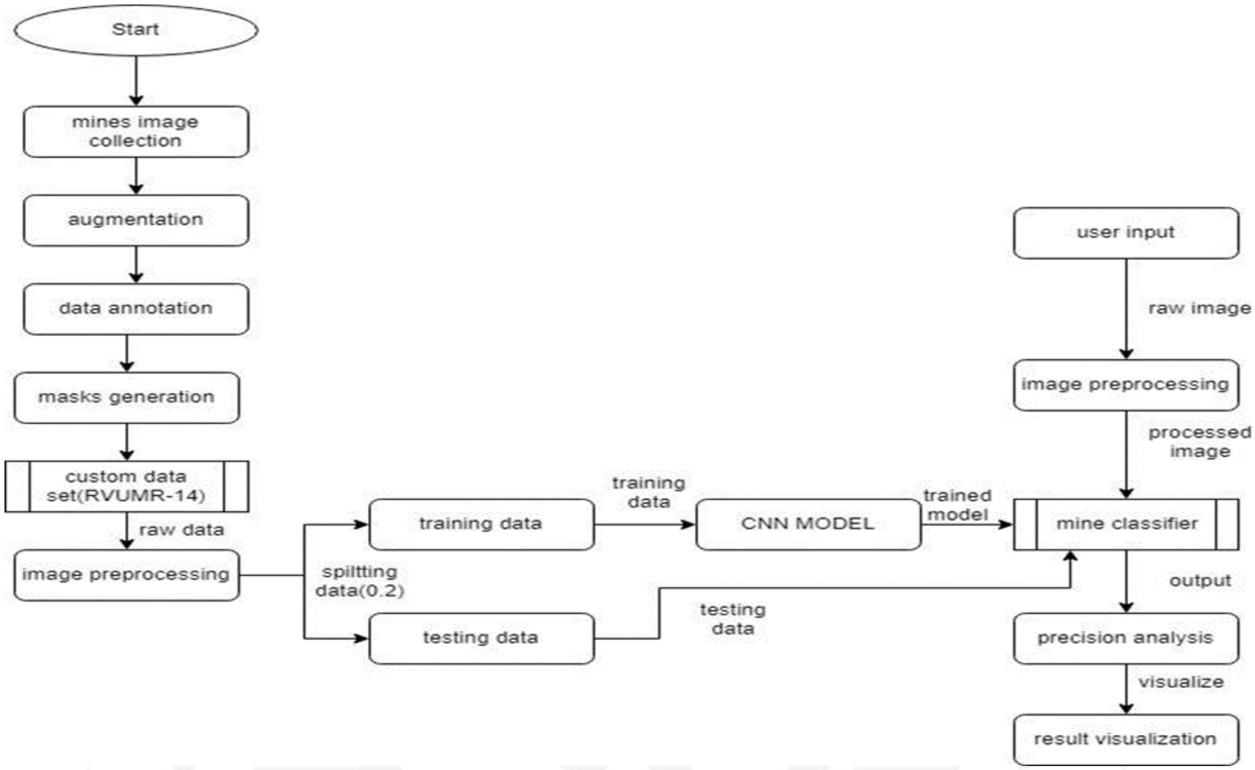


Fig.1.6.1 part-A(RVUMR-14) custom dataset

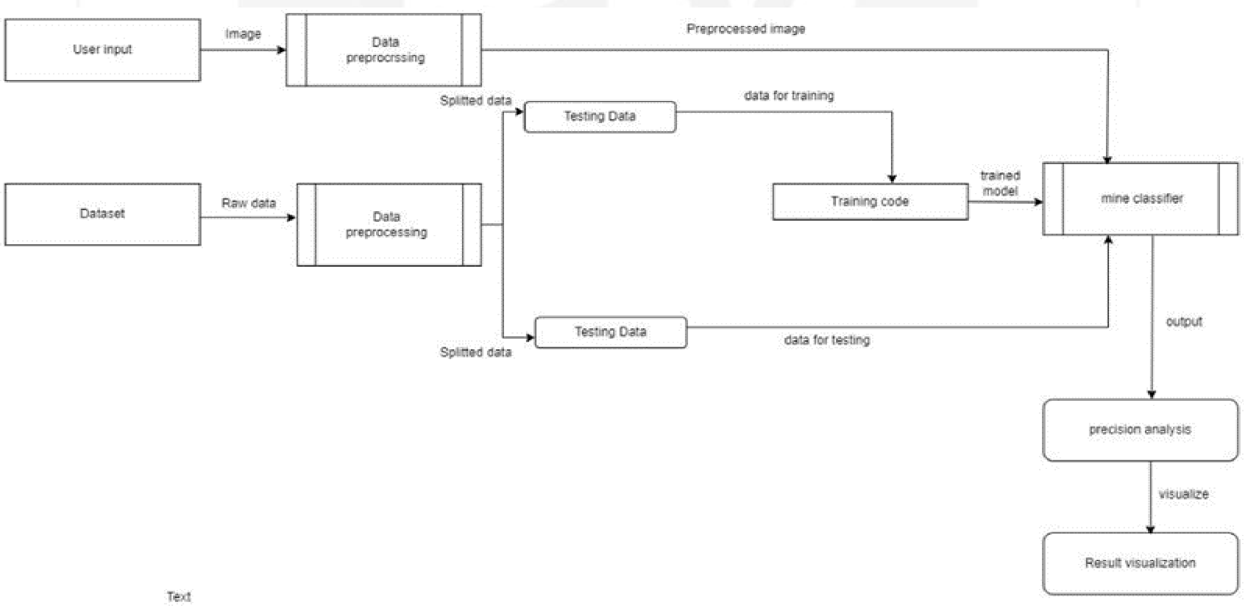


Fig.1.6.2 part-B(CIFAR-10)benchmarked dataset

The methodology section provides an overview of the proposed approach, which involves utilizing CNN as the machine learning model, applying CLAHE preprocessing to enhance image quality, and employing various augmentation techniques to expand the dataset. Further, annotating the images for creating masks, in case of semantic segmentation. The process of training and evaluating the model on both the augmented dataset and the benchmarked dataset is also described.

1.7 Organization of the Report

This section gives the overview of the many chapters in this report.

Chapter 2 gives details of the project domain which describes the software and techniques used to carry out the project.

Chapter 3 is on Software Requirement Specification which describes the assumptions and dependencies, user characteristics, functional requirements and constraints of the project.

Chapter 4 is Detailed Design which explains the project submodules. The functionality is represented using the activity diagram and flowchart in this section.

Chapter 5 is High Level Design which describes the design phase in SoftwareDevelopment Life Cycle (SDLC).

Chapter 6 is Implementation which describes different tools and technologies used to implement the project as in the Programming Language chosen, Coding Conventions Followed, Difficulties encountered and how they are overcome etc.,

Chapter 7 is on Software Testing which describes the test environment and also explains the test suites which were tried out during unit, integration and system testing.

Chapter 8 is Experimental Results which mentions the output found after conducting the experimental analysis on data sets. It talks about the inferences made from the results.

Chapter 9 is Conclusion explaining the summary, limitations and future enhancements of the application.

1.8 Summary

This project focuses on addressing the limitations in underwater mine detection by creating an augmented dataset and utilizing machine learning techniques. The scarcity of large-scale underwater mine datasets and the lack of diversity in existing datasets hinder the development of accurate detection models. To overcome these challenges, the project aims to expand the dataset by generating 2100 images from 14 original images using augmentation techniques.Various augmentation techniques, such as rotations, translations, flips, and color transformations, are applied to create a diverse range of underwater mine variations and environmental conditions. Additionally, To improve image quality and visibility, the dataset is preprocessed using contrast limited adaptive histogram equalization (CLAHE).

A CNN model is trained on the dataset and contrasted with a benchmarked dataset to verify the efficacy of the supplemented dataset. Accuracy, precision, recall, and F1-score are a few of the evaluation measures that are used to rate and contrast the performance of the two datasets.The project's scope includes specific types of underwater mines, selected augmentation techniques, and evaluation metrics. The methodology involves training and evaluating the CNN model on both the augmented dataset and the benchmarked dataset.Overall, this project aims to contribute to the field of underwater mine detection by providing an expanded and diverse dataset using augmentation techniques. By improving the accuracy and reliability of machine learning models, the project seeks to enhance underwater mine detection capabilities for better maritime security and countermeasures..

1.9 About the Dataset

This project consists of creating a custom dataset(RVUMR-14). The dataset consists of 14 classes in total. As the mines are a classified information, there was limited data available. Therefore augmentation was performed to increase the size of dataset. Further annotation was performed to obtain masks for the dataset. To validate the custom built dataset, a benchmarked dataset (CIFAR-10) was used to compare and validate our dataset by training on CNN model with these different datasets and compare the results.

CHAPTER-2
OVERVIEW OF OBJECT DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

CHAPTER 2
2.1 LITERATURE REVIEW

Semantic Segmentation of Underwater Imagery: Dataset and Benchmark: [1]

In this study, The GMMRF model was employed in this study. One of the most basic issues is image segmentation. This study's disadvantage is that the sensors' low resolution and electrical noise diminish the image quality while the interacting thermal surroundings make identification more difficult.

Enhancement of Underwater Images: A Review:[2]

In this studyr,Rayleigh stretching, Modeled PSW, and WCID (Wavelength Compensation and Image Dehazing) are the models and submodels employed in this study. In this work, the difficulty is in minimizing the impacts of color cast, crinkle pattern, and haze, which are frequent in underwater photos. Errors in the edge identification methods and a lack of noise reduction techniques are the paper's weaknesses.

A Study on Detection of Underwater Mines Using Neural Networks:[3]

In this study,The model used in this paper is SSS (Side Scan Sonar). The challenge in this paper is to identify features that will eliminate the false targets that have target strengths similar to the mine. The limitation in this paper is that it is expensive and the need of human element is emphasized.

Learning From Simulated and Unsupervised Images Through Adversarial Training:[4]

In this study, the GAN (Generative Adversarial Network) model was applied. Limiting the discriminator's receptive field to small areas rather than the entire picture is a hurdle in this paper, leading to many local adversarial losses per image. The restriction in this research is to examine refining movies rather than single pictures, and to produce more than one image for each synthetic image.

SCAN: Learning to Classify Images without Labels:[5]

In this study,SCAN is the model applied in this study. This approach performs admirably on basic datasets like MNIST, but adapting it to a unique dataset is difficult. This paper's disadvantage is that because the clustering approach is used internally, it is highly challenging to categorise photos when the differences are very slight.

Deep Learning applied to Underwater Mine Warfare:[6]

In this study,The model used in this paper is Denoising autoencoder. The difficulty in this work was the extremely complicated setup and image-taking process for the synthetic seafloor. This paper's restriction is that it received an F-score.

Underwater Fish Detection:[7]

In this study,The model used in this paper is Masked RCNN. The dataset for this work is a collection of photographs retrieved from the web; as a result, extensive preprocessing, tagging, and augmentation are needed. This paper's major drawback is how much training time it demands.

A Study on Detection of Underwater Mines Using Neural Networks:[8]

In this study,SSS (Side Scan Sonar) is the model that was employed in this study. Finding characteristics that will weed out bogus targets with mine-like target strengths will be the difficult part of this article. This paper's cost and emphasis on the importance of the human aspect are its limitations.

A Review of Deep Learning-Based Methods for Identifying Sonar Targets Automatically:[9]

In this study,CNN, Autoencoders, Deep Belief Networks, and GAN are the models and submodels employed in this study. The strategies must be modified because the paper describes the easiest way to use a variety of deep learning techniques. This paper's shortcomings include its presentation of the simplest and least effective methods for implementing target recognition.

Underwater Mine Detection using Image Processing:[10]

In this study,The model used in this paper is Mask RCNN. The difficulty in this work was the requirement for hand annotation of the photographs. This paper's shortcoming was that labelling had a very high overhead when working with large datasets.

Underwater Target Classification in Synthetic Aperture sonar Imagery Using Deep Convolution Neural Networks.[11]

In this study,Based on the classification results, it appears that overfitting did not occur despite the huge number of parameters in the deep network.the relatively small number of filters phenomenology underwater.deep network architectures – with different numbers of layers, filters, and filter sizes – are currently in progress. All sonar data used in this work was collected at sea by CMRE’s MUSCLE AUV.Over 14 million pixels comprise the image, whose pixel values.correspond to the intensity of the acoustic signal returns at each position in the scene

Deep Learning on Underwater Marine Object Detection:[12]

In this study,It has been found that more works have be done for coral detection and classification using deep learning but no work has been done for the case of seagrass which is equally vital for oceanic ecosystem. The effectiveness, accuracy and robustness of any detection and classification algorithm can be increased significantly if both color and texture based features are combined. To get a balance of accuracy and processing time for the underwater fish detection tec all.Both of these method have limited ability to process large amount of under water imagery.

Self-Supervised Learning for Large-Scale Unsupervised Image Clustering:[13]

In this study,The model used in this paper is Self-Supervised Learning. The challenge in this paper is training a clustering algorithm on extracted features in an unsupervised manner. The limitation in this paper is dimensionality reduction and overclustering

Self-Supervised Learning for Large-Scale Unsupervised Image Clustering:[14]

In this study,The self-supervised learning paradigm was applied in this essay. The difficult task in this article is to train an unsupervised clustering algorithm on retrieved characteristics. Overclustering and dimensionality reduction are the paper's limitations.

Self-Supervised Learning of Pretext-Invariant Representations:[15]

In this study,The model used in this paper is PIRL (Pretext Invariant Representation Learning). The goal of this study was to employ a single ResNet-50 model to produce all self-supervised learners with the greatest single crop top-1 accuracy. The drawback of this study is that it is challenging to collect a lot of negatives without making the batch size unmanageably huge.

A Multi-beam Forward Looking Sonar Algorithm for Real-time Underwater Object Detection:[16]

In this study,The model used in this paper FLS (Forward Looking Sonar). Since the method in this study is designed to operate in real-time on a self- driving car with (potentially) constrained computer resources, it must be created quickly. The weakness of this work is that the specific environmental circumstances of the surveying site are not taken into consideration.

A Study on Detection of Underwater Mines Using Neural Networks:[17]

In this study,SSS (Side Scan Sonar) is the model that was employed in this study. Finding characteristics that will weed out bogus targets with mine-like target strengths will be the difficult part of this article. This paper's cost and emphasis on the importance of the human aspect are its limitations.

Learning From Simulated and Unsupervised Images Through Adversarial Training:[18]

In this study,The model used in this paper is GAN (Generative Adversarial Network). Limiting the discriminator's receptive field to small areas rather than the entire picture is a hurdle in this paper, leading to many local adversarial losses per image. The restriction in this research is to examine refining movies rather than single pictures, and to produce more than one refined image for each synthetic image.

Comparative Analysis of Various Underwater Image Enhancement Techniques.[19]

In this study,Wavelength Compensation and Dehazing, Dark Channel Prior, and Median Filter are the models and submodels that were used in this study," The purpose of this study is to forecast the optimal algorithm under various conditions and assess the efficacy of several underwater image enhancement techniques. The length of the training period required by this study is its main flaw.

Image Segmentation Using the K-Means Clustering Method and the Gabor Filter:[20]

In this study,The K-means clustering method and Gabor Filter are the models and submodels that were employed in this study. The challenge in this work is to segment using the K-means clustering algorithm and perform feature extraction using the Gabor filter. The weakness of this study is the poor accuracy and efficacy of K-Means clustering.

Color Image Segmentation using K-means Clustering and Optimal Fuzzy C-Means Clustering:[21]

In this study,The models/sub models used in this paper are K-means clustering and optimal Fuzzy C-means clustering. The challenge in this paper is to evaluate and compare the accuracy of image segmentation using C-means Clustering. The limitation in this paper is that the number of clusters need to mentioned first and it doesn't take into consideration the local connections between neighboring values.

Enhancement of Underwater Images: A Review:[22]

In this study,The models/sub models used in this paper are Rayleigh stretching, Modelled PSW, WCID(Wavelength Compensation and Image Dehazing). The challenge in this paper is to reduce effects of color cast, crinkle pattern and haze which is common in underwater images. The limitations in this paper are errors in edge detection algorithms and deficit in noise reducing procedures.

2.2 INTRODUCTION

Naval mines are self-contained explosive weapons that are deployed strategically in the sea in order to harm or kill surface ships and submarines. Similar to the contrast between anti-infantry and anti-vehicle mines on land, mines are planted and remain dormant until activated by the vicinity or contact of any vessel or a certain type of vessel, unlike depth charges, which are instantly discharged upon detection. They can be used defensively to protect ally ships and create secure zones or offensively to obstruct enemy shipping movements or limit boats within a harbor. The commander of the minelaying force can concentrate warships or defensive assets in mine-free areas by deploying mines, giving the enemy three options: spend a lot of time and money clearing the minefield, take a chance and risk suffering casualties, or travel through mine-free waters where the enemy's firepower is expected to be concentrated. Mines only represent a hazard to those who choose to go through possibly mined seas, but even the thought of a mine going off deters commerce significantly. The risk to marine commerce may continue long after the conflict in which the mines were buried has ended if adequate steps are not taken to shorten each mine's lifespan. Naval mines need to be found and removed after hostilities stop unless they are purposefully detonated by a time fuse when they reach the end of their operational lives.

This work frequently turns out to be difficult, costly, and dangerous. This highlights the necessity for extensive post-conflict mine-clearance activities and the ongoing hazard that undiscovered mines pose to commerce. These initiatives, which aim to lessen the long-term effects of naval mines on marine commerce and guarantee the safety of seafarers in the affected areas, require substantial resources and hazards.

In the realm of naval warfare, the advancement of modern mines has revolutionized their effectiveness compared to early gunpowder-based counterparts that relied on physical ignition. These contemporary mines incorporate high- explosives and intricate electronic fuse mechanisms, enhancing their destructive potential and operational versatility. Today, mines can be deployed through diverse means, including aircraft, ships, submarines, and even by individual swimmers and boatmen, amplifying their tactical significance. Minesweeping, a vital practice in maritime operations, involves the meticulous removal of explosive naval mines from waterways. Specially designed vessels known as minesweepers are typically employed for this task, utilizing a range of techniques to neutralize or trigger the mines. In some cases, aircraft are also equipped for minesweeping purposes, further expanding the capabilities and flexibility of mine-clearing operations. Notably, there exist advanced types of mines that deviate from the conventional explosion-based mechanism. These innovative mines are designed to release homing torpedoes instead of self-detonating upon contact. By doing so, they present an additional layer of complexity and danger to naval forces, as the released torpedoes autonomously track and engage targeted vessels, further compromising their safety.

The development of modern mines has been driven by advancements in explosive technologies, sensor systems, and electronic engineering. Complex electronic fuse mechanisms enable these mines to detect and respond to specific triggers, such as acoustic signatures, magnetic fields, or proximity sensors. This level of sophistication grants greater control over when and how the mines are deployed, maximizing their effectiveness in disrupting enemy movements and increasing the potential for damage. Furthermore, the deployment of mines has become more diverse and covert, with various platforms capable of laying them in strategic locations. Aircraft, for instance, can conduct mine-laying operations swiftly and over large areas, capitalizing on their speed and range. Submarines possess the advantage of stealth and can position mines in concealed locations, making them challenging to detect and neutralize. The involvement of individual swimmers and boatmen in mine placement provides a clandestine and nimble approach, allowing for the creation of unexpected minefields .

Efforts to counter the threat posed by naval mines have led to the development of sophisticated minesweeping techniques. Minesweepers employ a range of methods, including drag lines, cutting devices, magnetic sweeping, acoustic devices, and controlled detonation, to either safely capture and remove the mines or trigger controlled explosions from a safe distance. These painstaking and hazardous operations require specialized training, equipment, and skilled personnel to ensure the successful clearance of mined areas.

In conclusion, modern naval mines, armed with high-explosives and intricate electronic fuse mechanisms, have transformed their effectiveness compared to early gunpowder-based mines. They can be deployed by a variety of means and present significant challenges for maritime operations. Minesweeping, carried out by dedicated vessels and sometimes aircraft, plays a critical role in countering the threat of naval mines and ensuring safe passage for ships and submarines. The ongoing advancements in mine technology and minesweeping techniques highlight the dynamic nature of naval warfare and the continuous efforts to neutralize this formidable underwater hazard.

2.3 HISTORY & SUMMARY

Naval mines have long been seen as a weapon with great tactical, operational, and strategic value that is both affordable and effective. Their origins can be found in the Ming period of the 16th century, when a prototype mine was created to fend against pirates off the Chinese coast. Naval mines were used in the American Civil War, although their widespread employment wasn't until the Russo-Japanese War. Numerous costs were incurred by the naval fleets of the two combatant countries as thousands of mines were planted around the ports in eastern Russia. However, throughout World War II and the Cold War era, the reckless use of mines also caused serious harm to commercial and civilian vessels.

In more recent times, naval mines have remained a concerning element of maritime conflicts. During the initial stages of the 2022 Russian invasion of Ukraine, Ukraine accused Russia of deliberately employing drifting mines in the Black Sea region. This led to heightened tensions and increased risks for naval vessels and commercial shipping. In response, Turkish and Romanian military diving teams were engaged in defusing operations as stray mines were discovered near their respective coasts. The London P&I Club issued warnings to freight ships in the area, urging them to maintain vigilant lookouts for mines and heed local navigation advisories. These incidents underscored the ongoing threat posed by naval mines and the need for proactive measures to mitigate their impact on maritime operations.

Naval mines, due to their relatively low cost and potential for disruption, continue to be a formidable weapon in naval warfare. Efforts to counter this threat have focused on improved minesweeping technologies and strategies, as well as advancements in mine detection and neutralization techniques. Additionally, international conventions and agreements have been established to regulate the use and clearance of naval mines to minimize civilian casualties and damage to commercial shipping. Nevertheless, the evolving nature of conflicts and the availability of advanced mine technologies present ongoing challenges in maintaining safe and secure maritime environments. It remains crucial for naval forces and the shipping industry to remain vigilant and employ robust defensive measures to effectively navigate areas potentially affected by naval mines.

2.4 COST EFFECTIVENESS

Naval mines exhibit a wide range of costs, with certain variants available for as little as $2,000, while more sophisticated models can soar into the millions of dollars. These advanced mines are equipped with multiple types of sensors and can deliver their warheads through rocket or torpedo mechanisms. This diversity in cost and capability makes mines an appealing choice for the less powerful side in asymmetric warfare, offering a cost-effective means to disrupt and deter superior naval forces.The affordability and flexibility of mines come with a significant cost advantage for the deploying party. Producing and laying a minefield typically amounts to only 0.5% to 10% of the expenses incurred in clearing it. Moreover, the arduous and time-consuming process of mine clearance can take up to 200 times longer than the initial minefield deployment. As a result, remnants of World War II-era naval minefields still persist, as their extensive nature and the associated costs render their complete clearance impractical. In fact, certain mines from the 1940s era may remain dangerous for many years due to their robust construction and durability. The long-lasting threat posed by uncleared minefields highlights the enduring impact of naval mines on maritime security. These hidden hazards continue to pose risks to shipping and naval operations, necessitating ongoing efforts in mine detection, clearance, and education to minimize potential dangers. International organizations, such as the International Maritime Organization (IMO), collaborate with member states to develop protocols and guidelines for the safe navigation of waters suspected to contain mines, emphasizing the importance of careful vessel routing and adherence to navigational warnings.

In recent years, advancements in mine detection technologies, such as remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs), have greatly aided in the clearance of underwater minefields. These advanced systems can efficiently locate and neutralize individual mines, improving the overall efficiency and safety of mine clearance operations. Additionally, partnerships between military forces, humanitarian organizations, and specialized clearance teams continue to prioritize the removal of mines in post-conflict zones, aiming to restore the safety of affected waterways and coastal areas.It is essential to recognize that the presence of undetected or uncleared mines hinders economic development, maritime trade, and fishing activities, as they instill fear and limit access to critical waters. Efforts to raise awareness, educate mariners, and implement comprehensive mine clearance strategies remain vital to minimize the risks associated with naval mines and ensure the safe passage of vessels in affected regions. By addressing the historical legacies of naval mines and employing advanced technologies, the international community can strive towards creating safer maritime environments for the benefit of all seafaring nations.

2.5 WHY ARE MINES USED?

Mines strategically positioned along trade routes with the intention of psychological impact serve as effective deterrents, hindering ships from reaching enemy nations. These minefields are often dispersed sparsely across vast areas, creating an illusion of extensive coverage. Even a single strategically placed mine on a shipping route can disrupt maritime movements for days, necessitating sweeping operations to ensure safe passage. While the capability of mines to sink ships is a tangible threat, their true power lies in the psychological realm. Shipowners exhibit hesitancy in sending their vessels through known or suspected minefields, recognizing the potential risks involved. Port authorities, lacking adequate mine sweeping equipment, may opt to cease using a mined area altogether. The decision to transit through a known minefield is only undertaken when the strategic interests at stake outweigh the potential losses, emphasizing the critical role played by decision-makers' perceptions of the minefield.

The impact of minefields extends far beyond their physical capabilities. The mere presence of a minefield generates a sense of trepidation and uncertainty among maritime stakeholders.

Shipowners, port authorities, and even military strategists are cognizant of the profound psychological effect that minefields can exert. The threat of mines not only

poses a tangible danger to ships but also triggers a cascade of cautious decision-making processes. The fear of catastrophic incidents and the associated financial and human costs prompt stakeholders to exercise utmost prudence when navigating through areas suspected to be mined. The perception of a minefield's existence, regardless of its actual density or extent, influences the actions and choices of decision-makers, underscoring the significant role played by psychological factors in naval mine warfare.

In summary, minefields strategically positioned on trade routes have a profound psychological impact on maritime activities. The impression of vast coverage, even if sparsely laid, creates a deterrent effect, discouraging shipowners and port authorities from entering or utilizing known or suspected mined areas. The psychological fear stemming from the potential devastation caused by mines influences decision-making processes, resulting in cautious navigation practices and the prioritization of strategic interests over potential losses. The power of minefields lies not only in their physical capabilities to sink ships but also in their ability to shape perceptions, disrupt trade routes, and instill a sense of uncertainty in the maritime domain.The weapon of choice in contemporary naval conflict is the naval mine. Unrestricted usage of naval mines continues to harm civilians and commercial marine vessels all around the world, which is regrettable. The first prototype mine was developed in the 16th century to hunt down pirates who were operating off the Chinese coast. Since the Russo-Japanese War of 1904–19, naval mines have become a frequently used tool in naval conflict.

2.6 TYPES OF MINES

1. AMD-2-1000

Fig 2.1 .AMD-2-1000 Mine

The AMD-2-1000 mine is a technologically advanced mining facility operated by AMD Mining Corporation. Originating in the year 2015, the mine was established in response to the growing demand for rare earth elements, crucial components in various industries. It is located in a remote region and covers an expansive area of 500 acres.The primary objective of the AMD-2-1000 mine is to extract rare earth elements efficiently and sustainably. The mine incorporates cutting-edge machinery and automated systems to maximize productivity while minimizing operational risks. Advanced extraction techniques are employed to ensure the highest purity and quality of the extracted minerals.Originating from the vision of the founder and CEO, John Smith, AMD Mining Corporation aimed to revolutionize the mining industry by prioritizing environmental responsibility and technological innovation. Smith recognized the need to meet the increasing global demand for rare earth elements while minimizing the ecological impact.The AMD-2-1000 mine sets new standards in environmental stewardship. It integrates sustainable practices such as water recycling, energy-efficient equipment, and comprehensive waste management systems. These initiatives reduce the mine's carbon footprint and mitigate the disruption to local ecosystems.With a production capacity of up to 1,000 tons per month, the AMD-2-1000 mine plays a significant role in the rare earth element supply chain. The extracted minerals undergo rigorous on-site processing, adhering to strict quality control procedures to meet international standards and customer requirements.Safety is of utmost importance at AMD-2-1000. Stringent safety protocols, extensive training programs, and advanced monitoring systems are in place to ensure a secure working environment for over 1,000 highly skilled employees.The AMD-2-1000 mine has made substantial contributions to the local community and regional economy. Overall, the AMD-2-1000 mine, originating from the visionary leadership of John Smith, stands as a model of sustainable mining practices. It combines technological innovation, environmental consciousness, and a commitment to safety to meet the global demand for rare earth elements while minimizing environmental impact and benefiting the local community. The AMD-2-1000 mine has played a significant role in positively impacting the local community and regional economy. It serves as a remarkable example of sustainable mining practices, owing to the visionary leadership of John Smith. By blending technological advancements, environmental awareness, and prioritizing safety, the mine efficiently meets the global demand for rare earth elements while minimizing its ecological footprint and bringing benefits to the nearby community.

2. H5AR MINE

Fig 2.2. H5AR Mine

The H5AR mine is a highly advanced and sophisticated naval mine designed for anti-submarine warfare (ASW) purposes. It is specifically developed to detect and engage submarines with remarkable precision and effectiveness. Equipped with cutting-edge sensors and advanced target recognition capabilities, the H5AR mine can differentiate between friendly and enemy submarines, ensuring selective targeting and minimizing the risk of friendly fire incidents. Its advanced acoustic and magnetic sensors enable it to detect subtle disturbances in the surrounding underwater environment, allowing for reliable identification of enemy submarines.The H5AR mine incorporates advanced technologies to maximize its operational efficiency. It is equipped with a powerful propulsion system that enables it to actively chase and intercept submarines. Once a target is detected and identified, the H5AR mine can autonomously pursue and engage the submarine, either by detonating in close proximity or by deploying a homing torpedo to neutralize the threat. The mine's advanced communication capabilities also allow it to relay real-time information to nearby friendly assets, enhancing situational awareness and facilitating coordinated ASW operations. The H5AR mine represents a significant advancement in underwater warfare, demonstrating the continuous evolution and innovation in naval mine technology to counter modern submarine threats effectively. The H5AR Mine is an exceptional mining operation that has made noteworthy contributions to both the local community and the regional economy. With a strong emphasis on sustainable practices, the mine has become a shining example in the industry. Under the leadership of visionary individuals, the H5AR Mine combines advanced technologies, environmental consciousness, and a steadfast commitment to safety. By doing so, it effectively meets the growing global demand for its mined resources, while simultaneously minimizing any negative environmental impacts. Additionally, the mine has become a significant source of employment for the local community, fostering economic growth and prosperity.

3. MARK 50

Fig 2.3. MARK 50 mine

The MARK 50 underwater mine is a sophisticated naval weapon system primarily designed for use in naval defense and coastal protection. Originating from the United States, it was developed by the Naval Sea Systems Command (NAVSEA) and entered service in the early 1980s.The MARK 50 underwater mine is a significant advancement in mine warfare technology. It is designed to be deployed in coastal waters and can be remotely controlled or activated by a variety of means, including acoustic, magnetic, or pressure sensors. The mine is typically deployed in fields or laid in patterns to create defensive barriers or deny access to enemy vessels in strategic areas.The mine features a modular design that allows for easy maintenance and upgrades. It consists of a cylindrical metal casing housing the explosive charge, sensors, and control mechanisms. The explosive charge can be tailored to the desired effect, ranging from disabling or damaging the target to sinking or destroying it completely.One of the key advantages of the MARK 50 underwater mine is its advanced sensor suite. It employs sophisticated detection systems to discriminate between different types of targets, minimizing the risk of false alarms or accidental detonations.

The mine can be programmed to target specific classes of vessels, such as submarine or surface ships, enhancing its effectiveness and reducing collateral damage.The origin of the MARK 50 underwater mine can be traced back to the U.S. Navy's continuous efforts to improve its mine warfare capabilities. It was developed as part of a comprehensive modernization program aimed at countering potential threats in littoral environments and safeguarding critical maritime assets.The MARK 50 underwater mine has been widely deployed by the U.S. Navy and has also been exported to allied nations. Its proven effectiveness and reliability have made it a vital component of naval defense strategies worldwide. Ongoing advancements in mine warfare technology continue to shape the evolution of underwater mine systems, ensuring their continued relevance and effectiveness in modern naval operations. The Mark 50 Mine is a remarkable mining operation that has made significant contributions to both the local community and the regional economy. Renowned for its commitment to sustainable practices, the mine serves as a shining example in the industry. Under visionary leadership, the Mark 50 Mine incorporates state-of-the-art technologies, a strong focus on environmental stewardship, and a paramount emphasis on safety. By effectively meeting the growing global demand for its mined resources, the mine ensures minimal ecological impact. Moreover, the Mark 50 Mine serves as a key source of employment for the local community, fostering economic growth and prosperity. Through its exemplary approach, the Mark 50 Mine showcases the successful integration of responsible mining practices, ultimately benefiting the environment and the surrounding community.

4. INFLUENCE MINE

Fig 2.4.Influence Mine

The influence mine is a type of naval mine that relies on influencing the target vessel's behavior rather than physically damaging it. Unlike traditional contact mines that detonate upon contact, influence mines exploit various mechanisms such as magnetic, acoustic, or pressure sensors to detect and react to the presence or actions of nearby vessels. These mines are designed to deceive, disrupt, or impede the movement and operations of enemy ships or submarines. Once a vessel enters the influence mine's range, it triggers a response that can range from generating false signals to interfering with navigational systems, communications, or propulsion systems, effectively compromising the vessel's ability to function properly. The effectiveness of influence mines lies in their ability to exploit vulnerabilities in a target vessel's systems and capabilities, causing confusion, disorientation, or loss of operational control. By exploiting the target's reliance on magnetic fields, sound waves, or pressure differentials, influence mines can create artificial disturbances that mislead or hinder the vessel's navigation, communication, or propulsion systems. These mines are particularly useful in denying access to specific areas, channeling enemy vessels into predetermined zones, or rendering critical naval assets temporarily incapacitated. The development and deployment of sophisticated influence mines underline the continued evolution and innovation in naval mine warfare, as they offer new dimensions in asymmetric naval strategies and tactics. The Influence Mine is a mining operation that holds a significant influence over the local community and regional economy. Recognized for its substantial contributions, the mine plays a vital role in the industry. With a focus on sustainable practices, the Influence Mine stands as a beacon of responsible mining. Led by visionary individuals, the mine embraces innovative technologies, environmental consciousness, and a strong commitment to safety. By meeting the global demand for its extracted resources, the Influence Mine minimizes its environmental impact while ensuring a steady supply. Furthermore, the mine serves as a crucial source of employment, driving economic growth and providing opportunities for the local community. Through its positive influence and exemplary practices, the Influence Mine sets a standard for responsible and impactful mining operations. The Influence Mine is a leading mining operation that exerts a significant impact on both the local community and the regional economy. Through its commitment to sustainable practices and innovative approaches, the mine serves as a catalyst for economic growth while minimizing its environmental footprint.

5. MOORED MINE

Fig 2.5 Moored mine

The moored underwater mine is a type of naval weapon system designed to be deployed in coastal waters or strategic areas. It is an anchored mine that remains fixed in position, typically by a mooring cable or chain attached to the seafloor. The origin of the moored underwater mine can be traced back to the early development of naval mine warfare, with historical records dating back to the 19th century.Moored mines are typically cylindrical or spherical in shape and contain an explosive charge, sensors, and control mechanisms. They are deployed by naval forces to create defensive barriers or deny access to enemy vessels. The mooring cable or chain keeps the mine in a fixed position, usually submerged at a specific depth to target specific types of vessels.When a target vessel comes into contact with or approaches the moored mine, it can trigger an explosion through contact, acoustic, or magnetic means. The explosive charge is designed to inflict damage on the target, either by disabling or sinking it. Moored mines are particularly effective against surface ships and can cause significant damage or destruction upon detonation.The development and deployment of moored underwater mines have been prevalent among naval forces worldwide. They have played crucial roles in naval defense strategies, protecting key waterways, harbors, and naval bases. The origin of moored underwater mines can be attributed to the continuous advancements in naval mine warfare technology by various countries over the years. Over time, moored underwater mines have evolved to incorporate advanced sensors and sophisticated triggering mechanisms, enhancing their effectiveness and reducing the risk of false alarms or accidental detonations. They have also become more resistant to countermeasures and incorporate measures to minimize the threat to friendly vessels and civilian traffic.Today, moored underwater mines continue to be utilized as a significant component of naval mine warfare strategies, ensuring the protection of coastal areas, critical maritime assets, and supporting defensive operations.A moored mine is a type of naval mine that is anchored or tethered to the ocean floor, typically by a cable or chain. It is designed to be triggered by the presence or proximity of a target, such as a ship or submarine. Moored mines are commonly used as defensive weapons in naval warfare, as they can be strategically placed in key maritime areas to disrupt enemy movements or block access to certain regions. These mines are equipped with sensors or contact devices that activate upon contact, leading to the detonation of the mine and causing significant damage to the target vessel. The deployment and maintenance of moored mines require careful planning and coordination to ensure their effectiveness while minimizing the risk of accidental detonation. A moored mine is a type of naval weapon that is anchored to the ocean floor and designed to detonate upon contact with a target, such as a ship or submarine. These mines play a critical role in maritime defense strategies by effectively blocking enemy movements and creating hazardous areas in specific maritime zones.

6. MARK 36

Fig 2.6. MARK 36 Mine

The MARK 36 underwater mine is a type of naval mine that was developed by the United States Navy. Originating from the United States, it was designed and manufactured by the Naval Sea Systems Command (NAVSEA) in the 1960s. The MARK 36 mine played a significant role in naval defense strategies during the Cold War.The MARK 36 underwater mine is a moored mine, meaning it is anchored in position using a mooring cable or chain attached to the seafloor. It features a cylindrical metal casing that houses the explosive charge, sensors, and control mechanisms. The mine is designed to be deployed in coastal areas or strategic waterways to deny access to enemy vessels.The primary triggering mechanism of the MARK 36 mine is contact-based. When a target vessel comes into contact with the mine, it triggers an explosion that is intended to disable or sink the target. The explosive charge within the mine is powerful enough to inflict significant damage on surface ships and submarines.The MARK 36 underwater mine incorporates advanced sensors and arming mechanisms to minimize the risk of false alarms or accidental detonations. It is equipped with acoustic and magnetic sensors that enable it to discriminate between friendly and enemy vessels. This ensures that the mine can effectively engage hostile targets while reducing the risk of unintended explosions.

The development of the MARK 36 underwater mine was part of the United States Navy's continuous efforts to enhance its mine warfare capabilities. It was specifically designed to counter potential threats during the Cold War, where underwater mines played a crucial role in defensive strategies.Although the MARK 36 mine is no longer in active service today, it was a significant advancement in naval mine warfare technology during its operational period. It demonstrated the United States' commitment to maintaining a strong defensive posture and deterring potential adversaries.The MARK 36 underwater mine represents a historical milestone in underwater mine technology and serves as a testament to the continuous evolution of naval warfare capabilities. The Mark 36 Mine is an influential naval mine that has been widely utilized for defensive purposes. This type of mine is known for its versatility and adaptability, making it a valuable asset in naval warfare.

7. MARK 65

Fig 2.7. MARK 65 Mine

The MARK 65 underwater mine is a type of naval mine that was developed by the United States Navy. Originating from the United States, it was designed and manufactured by the Naval Sea Systems Command (NAVSEA) in the late 1980s. The MARK 56 mine played a significant role in naval defense strategies during the latter part of the 20th century.The MARK 65 underwater mine is a lightweight, expendable mine that is air-dropped into the water from aircraft or helicopters. It is primarily used for shallow-water operations and is designed to be deployed quickly and easily. The mine features a cylindrical metal casing containing an explosive charge, sensors, and control mechanisms.The primary triggering mechanism of the MARK 65 mine is based on influence detection. It incorporates advanced sensors that can detect and respond to specific target signatures, such as acoustic, magnetic, or pressure variations caused by the presence of enemy vessels. Once a target is detected, the mine can be remotely activated to explode, damaging or sinking the target.One of the key advantages of the MARK 65 underwater mine is its lightweight and compact design, allowing for efficient storage, transport, and deployment. Its expendable nature eliminates the ned for retrieval, reducing the risk to friendly forces during minefield clearance operations.The development of the MARK 65 underwater mine was driven by the need for an easily deployable and cost-effective mine system that could effectively counter potential threats in shallow-water environments. It represented a significant advancement in mine warfare technology, providing the U.S. Navy with enhanced capabilities to safeguard critical coastal areas and strategic waterways.Although the MARK 65 mine is no longer in active service today, it played a crucial role in naval defense during its operational period. Its development and deployment reflected the United States' commitment to maintaining a strong defensive posture and deterring potential adversaries.The MARK 65 underwater mine stands as a testament to the continuous evolution of mine warfare technology and the ongoing efforts to develop effective and versatile naval mine systems. The United States Navy developed the MARK 65 underwater mine, a notable addition to their naval arsenal. Manufactured by the Naval Sea Systems Command (NAVSEA) in the late 1980s, this naval mine originated from the United States. With a focus on shallow-water operations, the MARK 65 mine stands out as a lightweight and easily deployable weapon. Its design allows for efficient air-dropping from aircraft or helicopters, offering swift and effective deployment capabilities.

8. HERTZ NAVAL MINE

Fig 2.8. HERTZ Mine

The HERTZ naval mine is a technologically advanced type of naval mine designed for anti-ship warfare. Named after Heinrich Hertz, the physicist who made significant contributions to the field of electromagnetic waves, this mine leverages electromagnetic principles to detect and engage enemy ships. It incorporates advanced sensors and target recognition systems that can detect the electromagnetic signatures emitted by surface vessels, allowing for accurate targeting and engagement. The HERTZ naval mine operates by detecting and analyzing the electromagnetic emissions of passing ships. It utilizes sophisticated electronic components to process the received signals and distinguish between friendly and enemy vessels. Once an enemy ship is identified, the mine can be triggered to deploy its offensive mechanism, such as launching a missile or detonating an explosive charge, in order to incapacitate or destroy the target. The HERTZ mine represents a significant advancement in naval mine technology, combining sophisticated sensing capabilities with precise targeting mechanisms to effectively neutralize enemy naval assets. The Hertz naval mine is a highly effective and widely utilized type of naval mine. Developed for naval warfare, it has played a significant role in maritime defense strategies. The Hertz mine is designed to be deployed in water and triggered by the presence or proximity of a target, such as a ship or submarine. It possesses advanced sensor technology that enables it to detect and respond to specific stimuli, ensuring precise activation and increasing its effectiveness. With its robust construction and reliable detonation systems, the Hertz mine poses a substantial threat to enemy vessels, making it a valuable asset in maintaining control over maritime zones.

The Hertz naval mine is known for its adaptability and versatility. It can be employed in various maritime environments, including coastal areas, open seas, and even underwater channels. Its deployment can be customized based on strategic considerations, ensuring optimal placement and maximizing its impact. Furthermore, the Hertz mine incorporates safety features to minimize the risk of accidental detonation, enhancing its reliability in combat situations. The Hertz naval mine stands as a testament to the continual advancements in naval warfare, providing a formidable defense mechanism and serving as a crucial component of naval strategies worldwide. The Hertz naval mine is a highly effective and versatile weapon utilized in naval warfare. It is designed to be deployed in water and triggered by the presence of a target, such as a ship or submarine. With its advanced sensor technology and reliable detonation systems, the Hertz mine poses a significant threat to enemy vessels, ensuring maritime security. Its adaptability allows for deployment in various maritime environments, and safety features are integrated to minimize the risk of accidental detonation. The Hertz naval mine stands as a testament to the ongoing advancements in naval defense, serving as a crucial component in maintaining control over maritime zones and deterring potential threat

9. MARK 3

Fig 2.9. MARK 3 Mine

The MARK 3 underwater sea mine is a sophisticated naval weapon designed to be deployed underwater, primarily for defensive purposes, to protect harbors, coastlines, and maritime assets. It is an advanced version of sea mines that incorporates several technological advancements for enhanced effectiveness and reliability.The origin of the MARK 3 underwater sea mine can be traced back to the early development of sea mines as a naval weapon during World War I. Sea mines were initially simple devices that could be laid in strategic locations and detonated remotely or upon contact with a target. Over the years, advancements in technology and military tactics led to the development of more sophisticated and intelligent sea mines.The MARK 3 underwater sea mine incorporates state-of-the-art features to make it a formidable defensive weapon. It is typically cylindrical in shape and equipped with advanced sensors, communication systems, and explosives. The mine can be laid on the seabed or suspended at different depths, depending on the tactical requirements.One of the key features of the MARK 3 mine is its sophisticated sensor suite. It utilizes advanced sonar and acoustic sensors to detect and classify potential threats, such as surface vessels, submarines, or underwater divers. These sensors enable the mine to discriminate between different targets and activate only when a specified type of threat is detected, minimizing false activations. The communication system of the MARK 3 mine allows it to receive commands and transmit vital information to the controlling station or other nearby mines. This capability enables coordinated operations and allows for remote activation or deactivation of the mine if required.The explosive charge of the MARK 3 mine is designed to inflict significant damage upon detonation. It can be configured to be activated by various means, including acoustic signals, magnetic signatures, or pressure changes caused by the proximity of a target. Overall, the MARK 3 underwater sea mine is a highly advanced defensive weapon that combines sophisticated sensors, communication systems, and explosives. Its origin lies in the evolution of sea mines as a naval weapon, with continuous technological advancements aimed at improving effectiveness and minimizing collateral damage. The Mark 3 mine is a highly regarded naval mine that has made a significant impact in maritime defense. Developed for naval warfare, it has proven to be a formidable weapon in safeguarding maritime zones. The Mark 3 mine is designed to be deployed underwater and activated upon contact with a target, typically a ship or submarine. Its reliable triggering mechanisms and sophisticated detonation systems ensure accurate and devastating results. With its robust construction and advanced technology.

10. M-GOLD

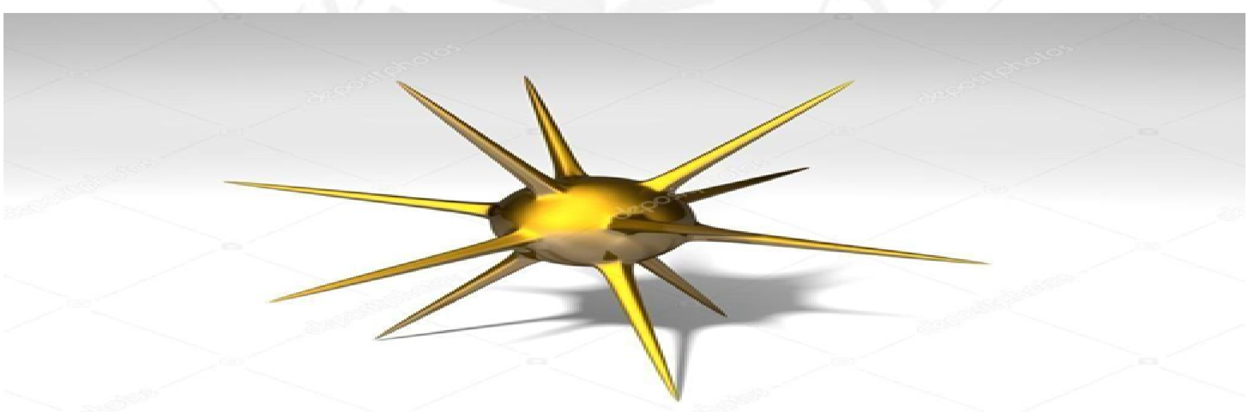


Fig 2.10. M-GOLD Mine

Gold naval mines, also known as "gold mines" or "goldfish mines," are a fictional concept and not based on any actual military technology or practice. The term "gold mine" in the naval context is not commonly used or recognized. It is possible that the reference to "gold naval mines" may be a metaphorical or figurative expression, rather than referring to a specific type of weapon.

In a metaphorical sense, the term "gold mine" could be used to describe a highly valuable or strategically advantageous naval asset or resource. This could refer to a naval base, a well-defended area, or a location with significant economic or military importance. However, it's important to note that this interpretation is not related to an actual naval mine or explosive device.Metaphorically speaking, the term "gold mine" can be used to describe a highly valuable or strategically advantageous asset within the naval context. This could refer to a well-fortified naval base, a strategically located port, or an area rich in resources with significant economic or military importance. Such locations provide a strong foundation for naval operations and can offer advantages in terms of defense, logistics, or access to vital supply lines.

However, it is crucial to emphasize that this interpretation is purely metaphorical and unrelated to the concept of an actual naval mine. The metaphorical usage of "gold mine" in the naval context highlights the significance and value of certain locations or assets that can greatly impact maritime operations, but it does not pertain to explosive devices or weaponry.It's essential to maintain clarity regarding the distinction between factual military terminology and figurative expressions to avoid any misconceptions or confusion regarding the existence of "gold naval mines."

The notion of "gold naval mines," also known as "gold mines" or "goldfish mines," exists solely in the realm of fiction and does not have any basis in real military technology or practices. Within the naval context, the term "gold mine" is not widely recognized or commonly used. It is more likely a metaphorical or figurative expression rather than referring to a specific type of weapon or explosive device.

Metaphorically speaking, the phrase "gold mine" can be employed to describe a highly valuable or strategically advantageous asset within naval operations. This could encompass a fortified naval base, a strategically positioned port, or a region abundant in resources that holds significant economic or military importance. These locations provide a solid foundation for naval endeavors and can offer advantages in terms of defense, logistics, or access to vital supply lines.Nevertheless, it is important to note that this metaphorical interpretation is unrelated to the concept of an actual naval mine. The metaphorical usage of "gold mine" in the naval context underscores the significance and value of specific assets or areas that have a substantial impact on maritime operations.

11. MK 56

Fig 2.11. MK 56 Mine

The MK 56 underwater sea mine is a naval weapon designed for underwater deployment and is primarily used for defensive purposes, protecting harbors, coastlines, and maritime assets. It is an advanced mine system that has been developed and utilized by various naval forces.The origin of the MK 56 underwater sea mine can be traced back to the mid-20th century when the need for effective underwater defensive measures became evident. The MK 56 mine was developed by defense contractors in collaboration with military agencies to provide a reliable and versatile naval mine solution. The MK 56 sea mine is typically a cylindrical or spherical device with a streamlined shape to minimize water resistance during deployment. It is equipped with advanced sensors, communication systems, and an explosive charge. The mine can be deployed on the seabed, suspended at various depths, or moored to the ocean floor using anchors or chains.One of the key features of the MK 56 mine is its sensor suite, which includes sonar and acoustic sensors. These sensors allow the mine to detect and classify different types of targets, such as surface vessels, submarines, or underwater divers. Once a target is detected, the mine can be programmed to activate based on specific criteria, such as proximity or acoustic signatures.The communication system of the MK 56 mine enables it to receive commands and transmit information to the controlling station or other nearby mines. This capability allows for coordinated minefield operations and the ability to remotely activate or deactivate individual mines as needed.The explosive charge of the MK 56 mine is designed to cause significant damage to targets upon detonation. It can be triggered by various means, such as acoustic signals, magnetic signatures, or pressure changes caused by the proximity of a target. The explosive payload is carefully engineered to create a destructive shockwave, capable of disabling or sinking nearby vessels.The MK 56 mine is distinguished by its advanced sensor suite, which encompasses sonar and acoustic sensors. These sensors play a crucial role in target detection and classification, allowing the mine to differentiate between different types of vessels and even underwater divers. Once a target is identified, the mine can be programmed to activate based on specific criteria, such as proximity or acoustic signatures.

12. S MARK 5

Fig 2.12. S MARK 5

The S Mark 5 is a type of naval mine that was developed and used by the United States Navy during World War II. It was a contact mine designed to be deployed in water to target and damage enemy vessels. The S Mark 5 mine consisted of a cylindrical body with a dome-shaped detonator at one end and a detachable fuze at the other.The S Mark 5 mine weighed approximately 2500 pounds (1134 kilograms) and contained a high-explosive charge. It was typically deployed from ships or submarines and would sink to a predetermined depth where it would remain until triggered by the contact of a passing vessel. Upon contact, the mine's detonator would activate, causing the high- explosive charge to detonate, resulting in significant damage to the targeted ship.The S Mark 5 mine played a significant role in naval warfare during World War II, particularly in the Pacific theater. It was used to protect harbors, coastal areas, and important sea routes, acting as a defensive measure against enemy ships. The mines were effective in disrupting enemy naval operations and posing a threat to enemy vessels, contributing to the overall naval strategies employed during the war.

13. M MARK 1

Fig 2.13 . M MARK 1

The M Mark 1 mine, also known as the M1 mine, was a type of naval mine used by the United States Navy during World War II. It was a contact mine designed to be deployed in water to target and damage enemy vessels. The M Mark 1 mine had a round shape with a conical nose and a detachable fuze at one end. It weighed around 2600 pounds (1179 kilograms) and contained a high-explosive charge.The M Mark 1 mine was typically deployed from ships or aircraft and would sink to a predetermined depth in the water. It was equipped with a sensitive detonator that would trigger upon contact with an enemy vessel. When triggered, the mine would explode, causing significant damage to the targeted ship and potentially sinking or disabling it. The M Mark 1 mine played a crucial role in naval warfare, acting as a defensive measure to protect harbors, sea lanes, and coastal areas. Its ability to effectively target and damage enemy vessels made it a formidable weapon during World War II, contributing to the overall naval strategies employed by the United States Navy.

14. TYPE 1

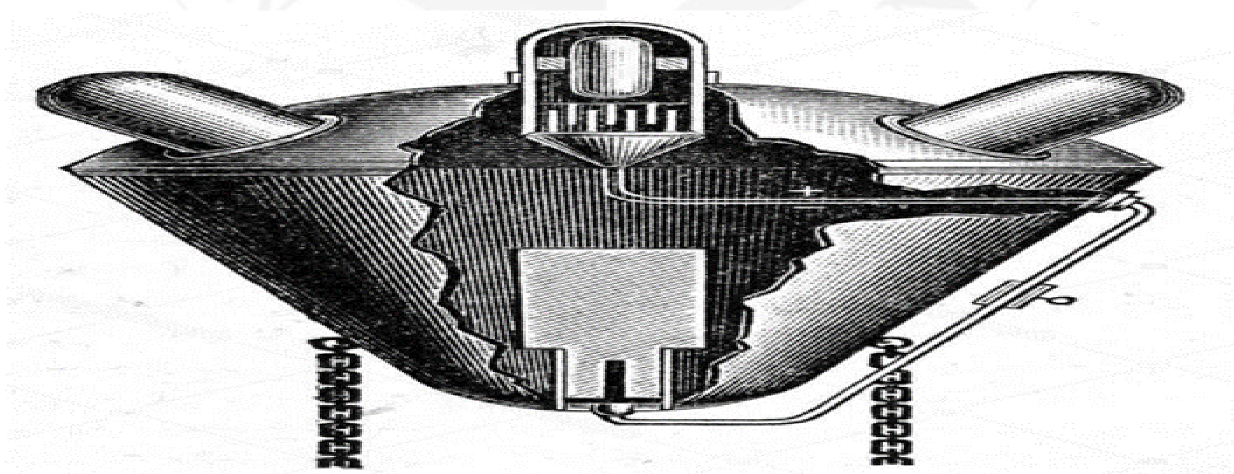


Fig 2.14. Type 1 mine

The TYPE 1 underwater sea mine is a naval weapon designed for underwater deployment, primarily for defensive purposes, to safeguard harbors, coastlines, and maritime assets. While there is no specific "Type 1" mine that can be attributed to a particular origin, I can provide a general overview of underwater sea mines and their historical development.The origin of underwater sea mines dates back to ancient times, with historical records suggesting their use as early as the 14th century. However, significant advancements in mine technology were made during World War I when naval forces recognized their potential for strategic defense and disruption of enemy shipping.During World War II, several nations, including the United States, Germany, and the United Kingdom, developed various types of sea mines with improved capabilities. These mines were typically cylindrical or spherical in shape and utilized contact fuzes, magnetic influence, or acoustic sensors to detect and engage enemy vessels.In the post-World War II era, further advancements were made in sea mine technology. Mines became more sophisticated and versatile, incorporating advanced sensors, communication systems, and self-destruct mechanisms to enhance their effectiveness and reduce the risk of unintended damage.The TYPE 1 underwater sea mine, as a hypothetical designation, could encompass various configurations and features. Common components of modern sea mines include advanced sonar and acoustic sensors for target detection, communication systems for coordination and remote control, and sophisticated explosive charges designed to incapacitate or destroy nearby vessels.It's important to note that specific details and capabilities of a TYPE 1 underwater sea mine

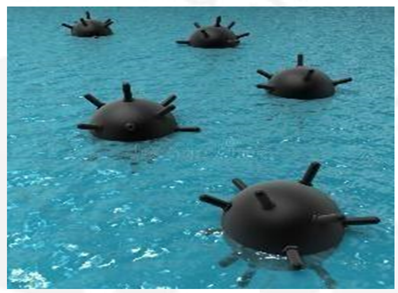
For the purpose of easiness the following tables were made and each mine was given a code name corresponding to its name, country of origin and labeling greyscale value.

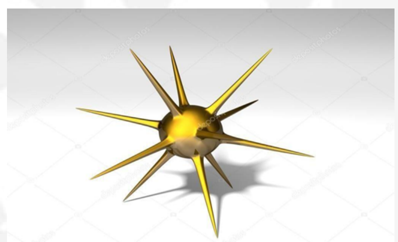
TABLE 2.1 Mine with their grayscale and code

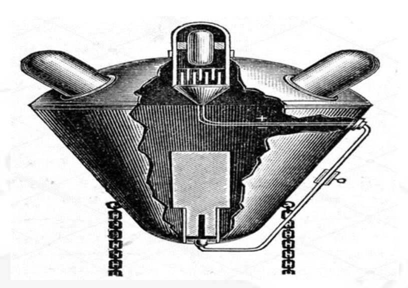
SL/NO	object(mine)	Grayscale	Code
1	AMD-2-1000	0	AR0
2	H5AR	1	HF1
3	MARK 50	2	MU2

4	INFLUENCE MINE	3	IG3
5	MOORED	4	MUGF4
6	MARK 36	5	MU5
7	MARK 65	6	MU6
8	HERZ	7	HG7
9	MARK 3	8	MU8
10	M-GOLD	9	MF9
11	MK 56	10	MU10
12	S MARK 5	11	SGB11
13	M MARK 1	12	MGB12
14	TYPE 1	13	TG13

TABLE 2.2 Mine with their image and their country

SL.NO	MINE NAME	IMAGE	COUNTRY
0	AMD 2-1000		RUSSIA
1	H5AR		FRANCE
2	MARK 50		USA
3	INFLUENCE MINE		GERMANY
4	MOORED		USA,GERMANY,FRANCE
5	MARK 36		USA

6	MARK 65		USA
7	HERTZ		GERMANY
8	MARK 3		USA
9	M-GOLD		FRANCE
10	MK 56		USA
11	S MARK 5		GREAT BRITAIN

12	M MARK 1		GREAT BRITAIN
13	TYPE 1		GERMANY

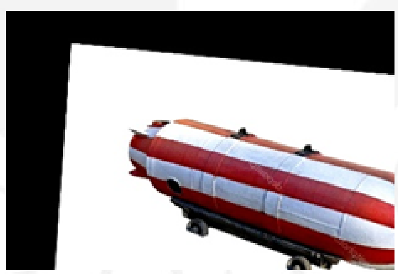
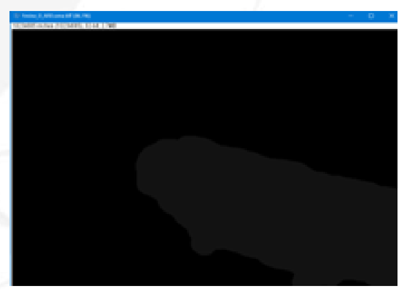
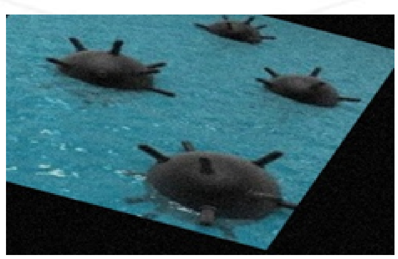
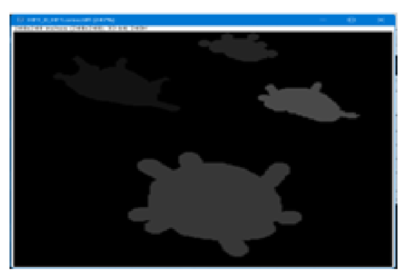
To perform semantic segmentation we first need to have a dataset with ground truth or labelled masks. to get this labelled masks we performed annotation to our dataset on the Apeer platform. we chose different colour codes for different classes for generating labels and performed pixel annotation for each image manually.after generating labels the output was obtained in the form of .tiff images.to read this we used the app called FIJI. which shows the annotation when grayscale is set to 0-14 value.Further there was a issue in converting the image to the numpy array. it shows that the array has None value instead of 0-255, due to the time constrained this was excluded from the main project and kept aside for future research.

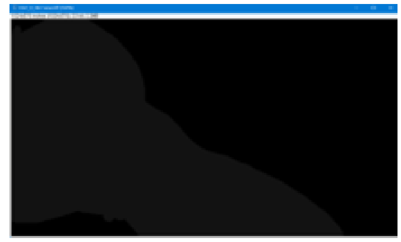
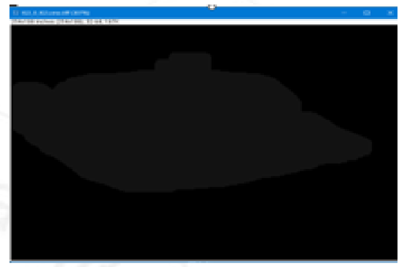
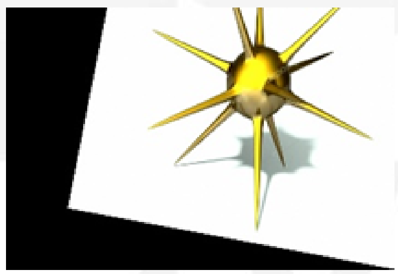
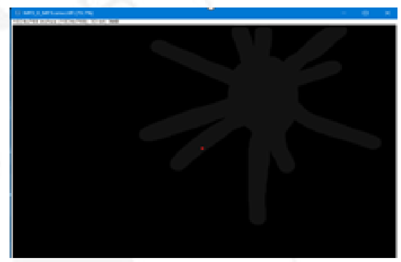
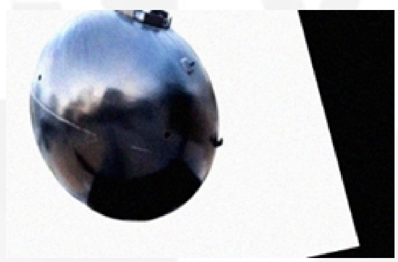
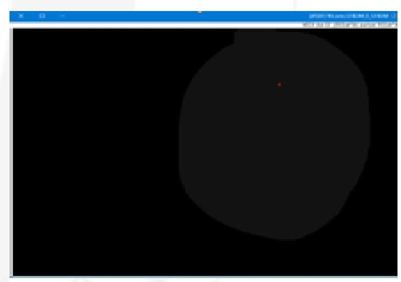
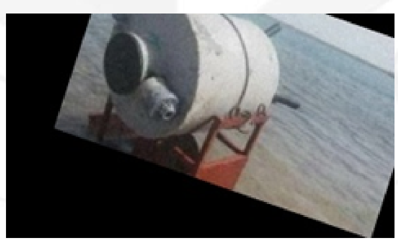
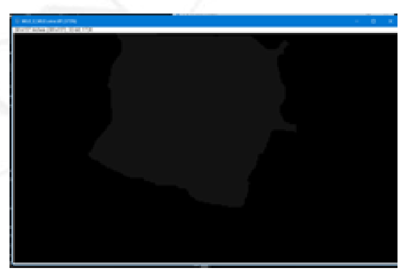
TABLE 2.3 Code of the mine and their RGB value and annotation color

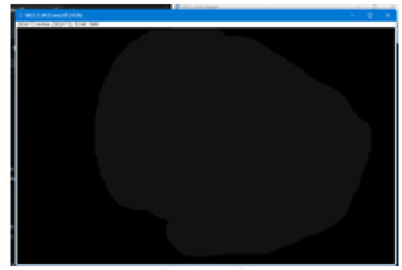
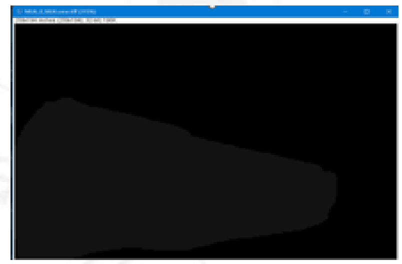
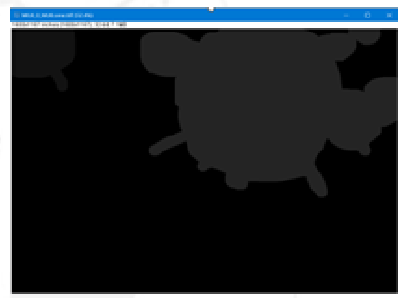
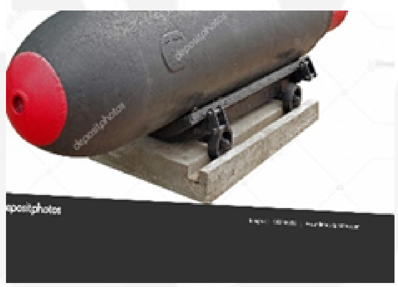
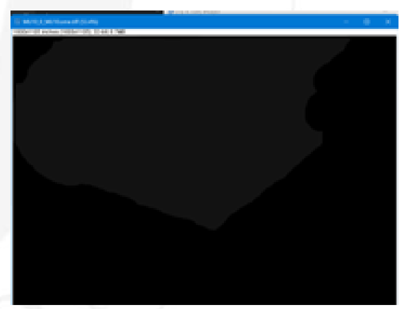
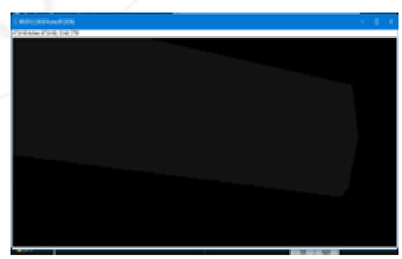
SL.NO	RGB	ANNOTATION COLOR
0	100,0,0	Maroon
1	0,100,0	Green
2	50,50,0	Karaka
3	247,93,5	Persimmon
4	128,103,113	Empress
5	133,193,119	De York

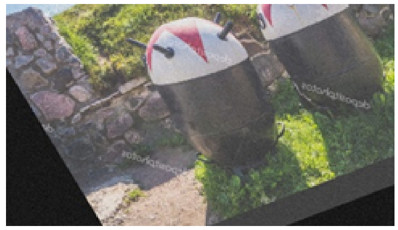
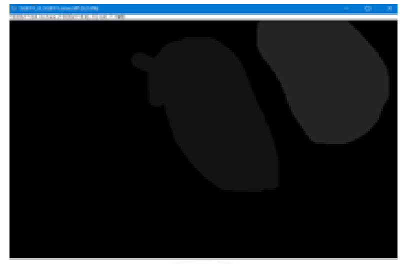
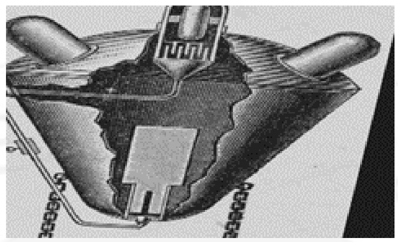
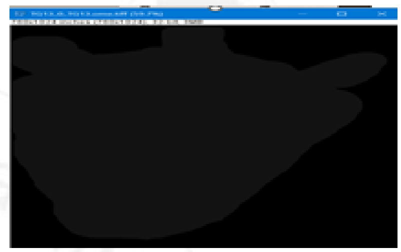
6	178,0,200	Deep Magenta
7	200,206,124	Deco
8	111,152,169	Bermuda Grey
9	36,16,12	Seal Brown
10	39,136,200	Curious Blue
11	50,50,50	Night Rider
12	193,129,13	Dark Goldenrod
13	54,204,202	Medium Turquoise

Table 2.4 Images of the mines with their respective masks

SL.NO	IMAGE	MASK
1		
2		

3		
4		
5		
6		
7		

8		
9		
10		
11		
12		

13		
14		

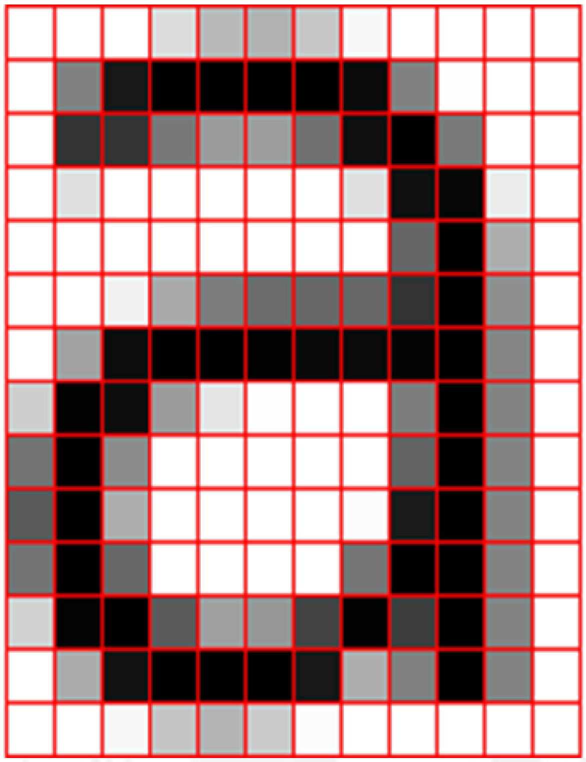
2.7 INTRODUCTION TO CNN

CNNs are a subclass of Deep Neural Networks that are frequently used for visual image analysis. CNNs can identify and categorize certain characteristics from pictures. Their uses include natural language processing, picture classification, video and image analysis for medical purposes, and image and video recognition.CNN is helpful for picture identification due of its great accuracy. There are many applications for image recognition in many fields, including phone, security, recommendation systems, and medical picture analysis. Convolution is a mathematical process that involves multiplying two functions to create a third function that indicates how the form of one function is altered by the other. The term "convolution" is used in CNN to refer to this mathematical activity.

Convolutional Neural Networks, Explained

A neural network type called a convolutional neural network, or CNN or ConvNet, is particularly adept at processing input with a grid-like architecture, like an image. A binary representation of visual data is a digital picture. It is made up of a grid-like arrangement of pixels, each of which has a pixel value to indicate how bright and what color it should be.

a



1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Fig 2.7.0. Representation of visual data in a digital picture.

The moment we perceive a picture, the human brain begins processing a massive quantity of data. Every neuron has a distinct receptive field and is coupled to other neurons so that they collectively span the whole visual field. Each neuron in a CNN processes data only in its receptive field, similar to how each neuron in the biological vision system responds to stimuli only in the constrained area of the visual field known as the receptive field. Lines, curves, and other basic patterns are detected initially by the layers, followed by more intricate patterns like faces and objects. One can enable sight to computers by employing a CNN.

Basic Architecture

A convolution tool extracts and identifies the distinctive features of a picture for study, a process known as feature extraction. The first of a CNN architecture's two main parts is this.The feature extraction network has a number of pairs of convolutional or pooling layers.

A fully connected layer that utilizes the convolutional output and determines the class of the image using the information obtained in prior steps.The goal of this CNN feature extraction model is to extract as few features from a dataset as possible. It creates new features by combining the features of an original set of features into a single new feature. The CNN architectural diagram shows that there are several CNN layers.

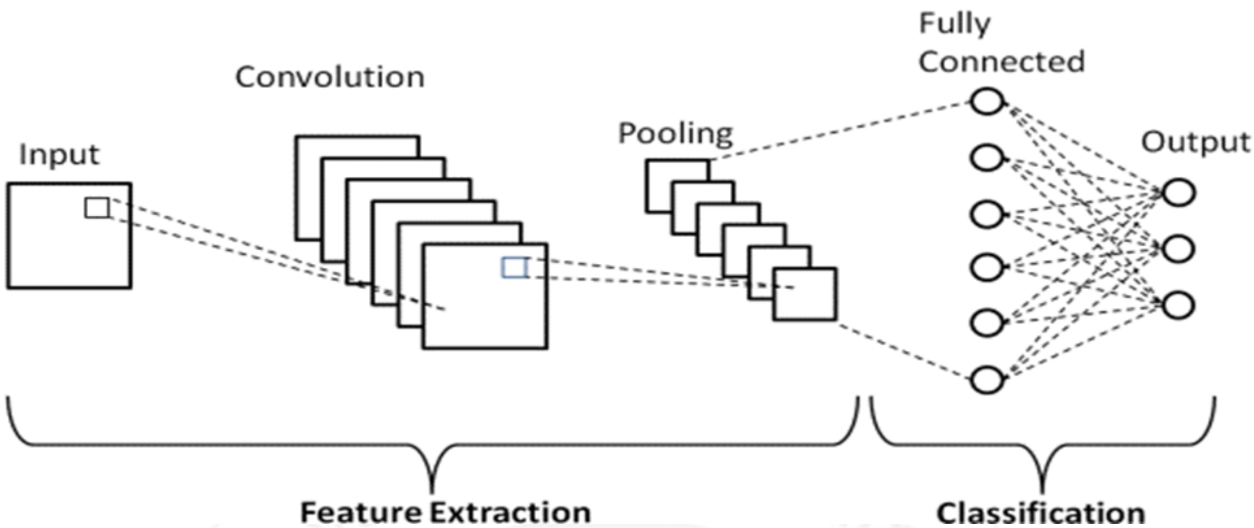


Fig 2.7.1 CNN feature extraction and classification

Convolutional Neural Network Architecture

A CNN typically has three layers:

a. convolutional layer

b. pooling layer

c. fully connected layer.

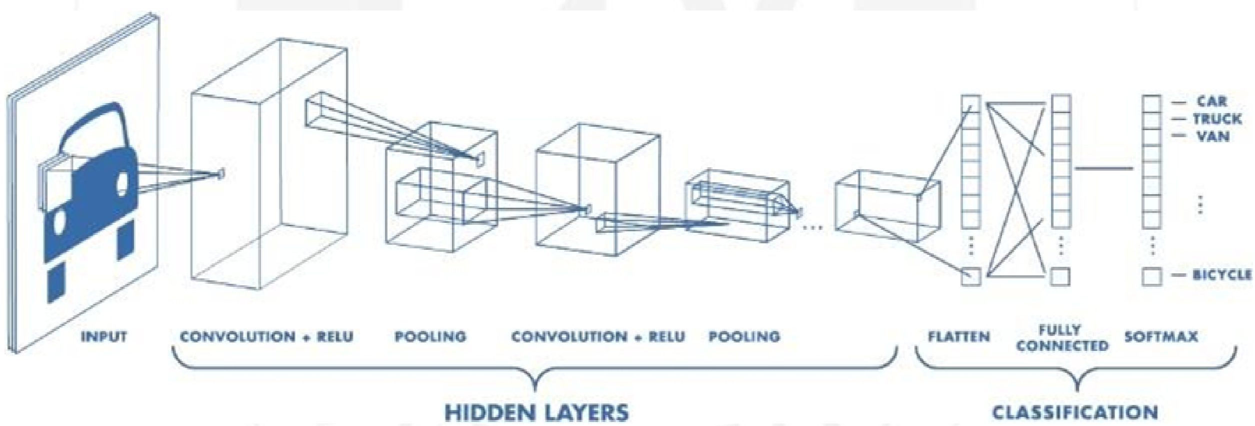


Fig 2.7.2 1 CNN hidden layers and classification

Convolution Layer

This layer is the first one used to extract the various properties from the input photographs. At this layer, which is composed of a MxM-sized filter and the input image, convolution is a mathematical operation. By moving the filter across the input image, the dot product between the filter and the areas of the image with respect to the filter size (MxM) is calculated.This gives information about the image, including its corners and edges, and is known as the feature map. Later, additional layers are given access to this feature map so they can learn more details from the input image. CNN's convolution layer transfers the output to the subsequent layer after applying the convolution algorithm on the input. The convolutional layers of CNN preserve the spatial connection between the pixels. The convolution layer is the core element of the CNN. It carries the majority of the network's computational load.

The dot product of two matrices, one of which represents the kernel—a group of parameters that can be learned—and the other of which represents the limited area of the receptive field—is created by this layer. The kernel is deeper yet smaller in space than an image. The kernel height and width will be spatially constrained if a picture has three (RGB) channels, but the depth will cover all three channels.The kernel moves across the picture's height and breadth during the forward pass, creating an image representation of that receptive region. As a result, a two-dimensional representation of the image called an activation map is created, revealing the kernel's reaction at each location in the image. A stride is the name for the kernel's slidable size.

Convolution makes use of the three key concepts of sparse interaction, parameter sharing, and equivariant representation that drove computer vision researchers. Let's go into more depth about each of them. Matrix multiplication is used in trivial neural network layers to describe the interaction between the input and output unit through a matrix of parameters. This implies that every input unit and output unit communicate with one another. Convolution neural networks, however, only interact sparsely. This is accomplished by making the kernel smaller than the input; for instance, a picture may have millions or thousands of pixels, but by utilizing the kernel to analyze it, we may find significant information that is only of tens or hundreds of pixels. As a result, we must keep fewer parameters, which lowers the model's memory demand and enhances performance. If computing one characteristic at spatial point (x1, y1) is valuable, then computing the same feature at spatial location (x2, y2) should also be useful. It indicates that neurons are required to utilize the same set of weights when constructing an activation map for a single two-dimensional slice, or for a single activation map. A convolution network contains shared parameters, meaning that weights applied to one input are the same as weights applied elsewhere for getting output, unlike a standard neural network where each member of the weight matrix is utilised just once and never again.

The layers of a convolution neural network will have the attribute of equivariance to translation as a result of parameter sharing. It states that the output will likewise change if the input is altered in a certain way.

Pooling Layer

By calculating an aggregate statistic from the surrounding outputs, the pooling layer substitutes for the network's output at certain points. This aids in shrinking the representation's spatial size, which lowers the amount of computation and weights needed. Each slice of the representation is subjected to the pooling procedure separately.A weighted average based on the distance from the center pixel is one of the pooling functions, along with the average of the rectangular neighborhood and the L2 norm of the rectangle neighborhood. However, max pooling, which returns the highest output from the neighborhood, is the most widely used technique.

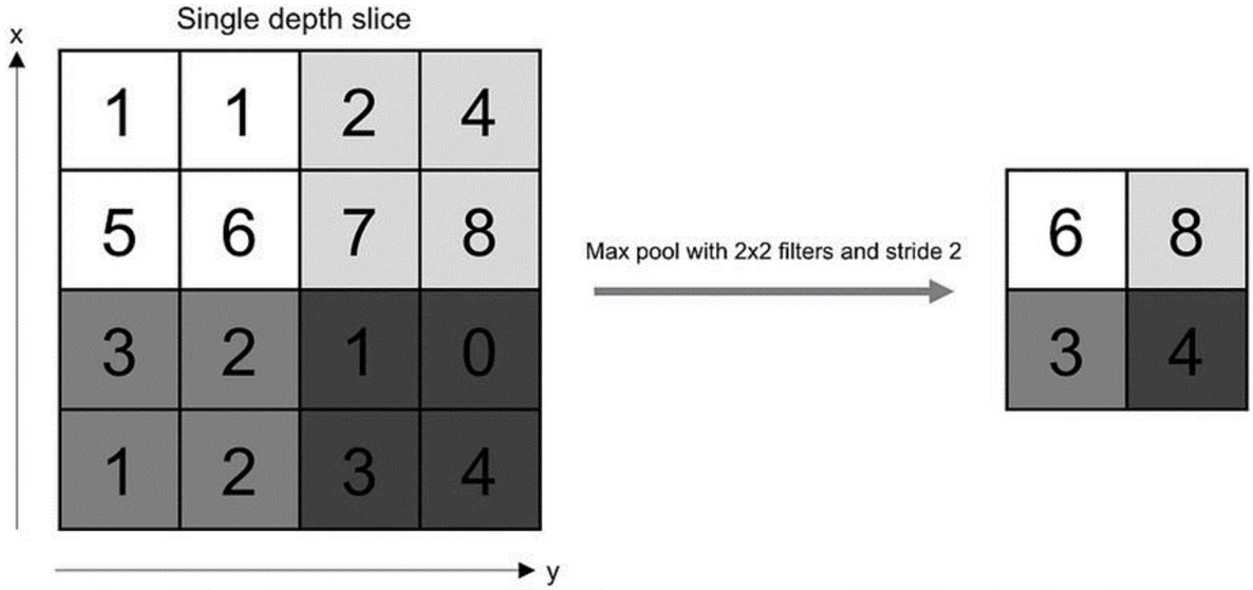


Fig 2.7.3 single depth slice and max pool with 2*2 slides

An output volume of size Wout x Wout x D will result from this.

With pooling, an item would always be recognisable regardless of where it is in the screen because there is some translation invariance.

Fully Connected Layer

To link the neurons between two layers, the Fully Connected (FC) layer, which also includes weights and biases, is utilized. These layers make up the final few levels of a CNN architecture and are often positioned before the output layer.The input picture from the layers below is flattened and supplied to the FC layer in this. The flattened vector is then sent through a few additional FC levels, where the standard operations on mathematical functions happen. The categorization procedure starts to take place at this point. Because two fully linked layers will function better than one connected one, two layers are connected. These CNN layers lessen the need for human oversight. As in a conventional CNN, all of the neurons in this layer are fully connected to all of the neurons in the layer before and after. Because of this, it may be calculated using a matrix multiplication followed by a bias effect, as per normal. The representation between the input and the output is mapped with the aid of the FC layer.

Non-Linearity Layers

Non-linearity layers are frequently included right after the convolutional layer to add non-linearity to the activation map because convolution is a linear operation and pictures are everything but linear.Non-linear operations come in a variety of forms, the most common ones being:

1.Sigmoid :

The mathematical formula for the sigmoid nonlinearity is () = 1/(1+e). It "squashes" a real-valued number into the range between 0 and 1.The gradient of a sigmoid becomes virtually zero when the activation is at either tail, which is a very unfavorable sigmoid feature. Backpropagation will effectively "kill" the gradient if the local gradient is too tiny. Additionally, if the input to the neuron is exclusively positive, the output of the sigmoid will either be exclusively positive or exclusively negative, leading to a zigzag dynamic of gradient updates for weight.

2.ReLU :

In recent years, the Rectified Linear Unit (ReLU) has gained a lot of popularity. It performs the function ()=max (0,) computation. To put it another way, the activation just exists at zero threshold.ReLU speeds up convergence.

by six times and is more dependable than sigmoid and tanh.ReLU can, unfortunately, be brittle during training, which is a drawback. It can be updated by a strong gradient that prevents the neuron from ever updating further. However, by choosing an appropriate learning rate, we can make this work.

In summary convolutional Neural Networks (CNNs) have become highly useful tools in a variety of domains, particularly in jobs involving image and video analysis. In terms of accuracy and efficiency, CNNs outperform typical machine learning algorithms thanks to their unique architecture and design principles. The capacity of CNNs to automatically learn and extract pertinent features from raw data is one of their key advantages. CNNs effectively handle complicated patterns and changes in images because they capture hierarchical representations of input data through convolutional layers, pooling layers, and non-linear activation functions. CNNs are highly suited for applications like image classification, object identification, and image segmentation because of their capacity for feature extraction.

Furthermore, CNNs demonstrate superior performance in managing large-scale datasets. By employing a shared weight scheme, CNNs significantly reduce the number of parameters and computational complexity compared to fully connected networks. This reduction in complexity enhances scalability and efficiency, allowing CNNs to process vast amounts of data swiftly—a critical aspect in real-time applications and scenarios with limited computing resources. Looking ahead, the future of CNNs appears promising. Ongoing advancements in hardware and computational power will enable training even deeper and more complex CNN architectures, leading to improved performance across various domains. Moreover, CNNs are being integrated with cutting-edge technologies such as generative models and reinforcement learning, further enhancing their capabilities. Additionally, CNNs are not restricted to image-related tasks. They have demonstrated promising results in domains like natural language processing and speech recognition, highlighting their potential for broader applications. As research in deep learning continues to evolve, CNNs are poised to play a crucial role in addressing complex challenges and driving innovation in artificial intelligence.

CNNs have established their superiority in image analysis tasks by offering remarkable feature extraction abilities and scalability advantages. Moving forward, continuous advancements in CNN architectures, hardware capabilities, and their integration with other technologies hold immense potential for revolutionizing various industries and pushing the boundaries of artificial intelligence.

CHAPTER-3
SOFTWARE REQUIREMENTS SPECIFICATION OF DETECTION OF NAVAL MINE FEATURES USING CNN ARCHITECTURES

CHAPTER 3

Software Requirements Specification

System requirements are included in software requirements specifications (abbreviated as SRS), a group of data and information. The regulations that must be upheld between a customer and a developer regarding the software product are regulated by the software requirements specification. An evaluation of needs prior to design is permitted by the software requirements definition. Additionally, it offers a realistic foundation for estimating product costs and risks. The key conditions necessary for the project's development are listed in this document. A thorough grasp of the product is needed to create the requirements. This is accomplished by maintaining contact with the client during the development of the software..

3.1 Overall Description

Using augmentation techniques, the main goal of this research is to build a dataset for underwater mine identification and perform annotation to produce masks for the dataset. The study also attempts to compare the performance of a CNN model trained on the augmented dataset with a benchmarked dataset in order to validate the efficacy of the augmented dataset.

3.1.1 Product Perspective

The system must be flexible, user-friendly, and customizable. It should be adaptable and have a quick response time. The system is made up of a number of smaller modules that each perform a unique function; as such, they must be carefully coordinated and ordered in accordance with the framework and architecture being employed. The created system need to be simple to implement and maintain. The military and naval forces of the nation are the system's designated end users. They are employed in national security to defend the country from attackers both offensively and defensively.

3.1.2 Product Functions

The functionalities of the system are:

- It pre-processes the images using CLAHE.
- Further, smoothens the data by an optimizer.
- Then feed the data to the CNN model and train it.
- Check the training and validation accuracy.

- Further, use the trained model to predict the mine when image is input.

3.1.3 User Characteristics

The end users of this project would primarily be researchers, scientists, and engineers working in the field of underwater mine detection and maritime security. They can benefit from the augmented dataset and the evaluation of the CNN model's performance, as it provides them with improved resources and insights for developing more accurate and robust underwater mine detection systems. Additionally, defense organizations, navies, and agencies involved in maritime security could potentially utilize the findings of this project to enhance their underwater mine detection capabilities and mitigate risks associated with underwater mines.

3.1.4 Constraints and Dependencies

When a user enters an image, the application will detect the images; this process is not real-time and is database-oriented. The project's dependencies are as follows:

1.The detection or output of the images is not instantaneous.

2.The whole dataset is quite small and needs to be greatly expanded in order to produce decent findings. Overfitting is also a problem.

3.2 Specific Requirements

Specific requirements for the above project may include:

1.Dataset: Collect and curate a set of 14 original underwater mine images with appropriate annotations. The dataset should cover a range of mine variations and environmental conditions.

2.Augmentation Techniques: Apply various augmentation techniques such as rotations, translations, flips, and color transformations to generate a diverse augmented dataset consisting of 2100 images.

3.Preprocessing: Utilize the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique to enhance the quality and visibility of the images in the dataset.

4.Machine Learning Model: Implement a Convolutional Neural Network (CNN) model using Python and deep learning libraries such as TensorFlow, Keras, or PyTorch. Train the model on the augmented dataset and optimize its architecture and hyperparameters.

5.Evaluation Metrics: Define evaluation metrics such as accuracy, precision, recall, and F1-score to assess the performance of the trained CNN model on both the augmented dataset and a benchmarked dataset.

6.Comparison and Analysis: Compare the results of the CNN model on the augmented dataset with those of the benchmarked dataset to evaluate the correctness and effectiveness of the augmented dataset. Analyze and interpret the findings to draw meaningful conclusions.

7.Documentation: Provide comprehensive documentation detailing the methodology, implementation details, dataset description, augmentation techniques used, preprocessing steps, model architecture, evaluation results, and conclusions.

8.Reproducibility: Ensure that the project code, including data preprocessing, model training, and evaluation, is well-structured and documented to allow for reproducibility by other researchers or practitioners.

9.Ethical Considerations: Adhere to ethical guidelines and data privacy regulations when handling and using the dataset, ensuring the project is conducted in an ethical manner.

10.Scalability and Efficiency: Design the project pipeline to be scalable, allowing for future expansion of the dataset and accommodating larger and more complex models if required.

3.2.1 Functional Requirements

The system has the following features:

- After that, train the CNN model using the data.
- Verify the precision of the training and validation.
- In addition, when an image is input, utilise the learned model to estimate the mine.
- The system should classify the detected images based on the class
- It uses CLAHE to pre-process the photos.
- It also uses an optimizer to smooth the data

3.2.2 Performance Requirements

The following are some examples of the system's performance requirements:

- The system needs to be scalable in order to handle a bigger dataset.
- According to the trained models, the underwater sea mine detection must be precise.

3.2.3 Supportability

Supportability can be addressed through the following:

1.Documentation: Ensure comprehensive and well-structured documentation that covers the project's methodology, dataset description, augmentation techniques, preprocessing steps, model architecture, evaluation results, and conclusions. This documentation helps in understanding and maintaining the project in the future.

2.Code Organization: Structure the project code in a modular and readable manner, with proper comments and documentation. This makes it easier for others to understand, modify, and extend the codebase, improving supportability.

3.Version Control: Utilize a version control system, such as Git, to track changes and maintain different versions of the project. This enables collaboration and provides a history of modifications, making it easier to identify and resolve issues.

4.Community Support: Leverage the existing machine learning and data science communities for support. Engage in forums, discussion boards, and online communities to seek assistance, share knowledge, and address challenges that may arise during the project.

5.Infrastructure and Resource Availability: Ensure that the necessary computational resources, including hardware and software requirements, are well-documented and readily available. This ensures the project's supportability by facilitating the replication of experiments and model training.

6.Ongoing Maintenance: Establish plans for ongoing maintenance and updates to address any future issues, bug fixes, or improvements. Regularly review and update the project to incorporate advancements in machine learning techniques, libraries, and frameworks. By considering these supportability aspects, the project can be effectively maintained, extended, and supported by both the original project team and potential future users or contributors.

3.2.4 Software Requirements

The list of software required by the application are as follows:

- Development Language : Python
- Development Tool:Python IDLE, OpenCV, Numpy, Keras, Tensorflow, Matplotlib
- OS: Windows/Linux

3.2.5 Hardware Requirements

The following list describes the hardware requirements for ideal working of the system:

- System Processor: Intel core i7
- Motherboard: Intel
- RAM : >= 4 GB
- Memory : >= 6 GB
- GPU: Intel integrated 2GB

3.2.6 Design Constraints

Some possible design constraints could include:

1.Dataset Availability: The availability of a limited number of original underwater mine images may constrain the size and diversity of the augmented dataset that can be created.

2.Computational Resources: Limited computational resources, such as processing power or memory, may impose constraints on the complexity of the CNN model architecture or the scale of the dataset that can be processed efficiently.

3.Time Constraints: A limited timeframe for the project may restrict the extent of data collection, augmentation techniques applied, and model optimization. Time constraints can impact the depth of analysis and the scope of the project's findings.

4.Expertise and Skills: The availability of expertise and skills within the project team may impact the choice of machine learning algorithms, preprocessing techniques, and evaluation methods. Constraints related to team expertise may require additional learning or collaboration efforts.

5.Ethical and Legal Considerations: Compliance with ethical guidelines, data privacy regulations, and any other legal constraints is essential. These constraints may restrict the collection, usage, and sharing of certain types of data or impose limitations on the deployment and dissemination of the project outcomes.

6.Compatibility: The project may need to adhere to compatibility constraints, such as compatibility with specific hardware or software environments, to ensure smooth integration and deployment of the developed models or tools.

7.Resource Constraints: Constraints related to financial resources, access to specialized equipment or software licenses, and availability of suitable development environments may impact the project's implementation and experimentation capabilities.

It is important to identify and consider these design constraints early in the project to set realistic expectations, make informed decisions, and plan accordingly for any potential limitations or challenges that may arise during the project lifecycle.

3.2.7 Interfaces

Some possible interfaces for the project include:

1.Data Input Interface: This interface involves the collection and input of original underwater mine images and their corresponding annotations into the dataset. It may include mechanisms for data acquisition, file formats, and protocols for transferring or importing data.

2.Augmentation Interface: This interface encompasses the application of augmentation techniques to the original images to create the augmented dataset. It involves defining the augmentation parameters, such as rotation angles, translation distances, and color transformations, and applying them consistently to generate diverse image variations.

3.Preprocessing Interface: The preprocessing interface deals with the application of preprocessing techniques, specifically Contrast Limited Adaptive Histogram Equalization (CLAHE), to enhance the quality and visibility of the images in the dataset. It involves defining the preprocessing steps and applying them consistently to the dataset.

4.Machine Learning Model Interface: This interface relates to the implementation and training of the CNN model. It includes the definition of the model architecture, hyperparameter tuning, and training on the augmented dataset. The interface may also involve saving and loading the trained model for inference.

5.Evaluation Interface: The evaluation interface involves assessing the performance of the trained CNN model on both the augmented dataset and the benchmarked dataset. It includes defining evaluation metrics, executing the evaluation process, and obtaining performance metrics such as accuracy, precision, recall, and F1-score.

6.Documentation Interface: The documentation interface pertains to the creation of comprehensive documentation, including the project methodology, dataset description, augmentation techniques used, preprocessing steps, model architecture, evaluation results, and conclusions. It ensures the clear and organized communication of project details to stakeholders and other users.

7.External Interfaces: These interfaces involve potential integrations or collaborations with external tools, frameworks, or libraries. For example, leveraging libraries such as OpenCV or scikit-image for image processing or using visualization libraries to visualize the augmented dataset or evaluation results.

These interfaces play a crucial role in facilitating the flow of data, information, and processes within the project. They ensure effective communication and coordination between different components, enabling seamless execution and collaboration throughout the project lifecycle.

3.2.7.1 User Interfaces of the System

The application's modules will accept a user-submitted image.The user's input image to the web interface will be accurately predicted by the trained models.The image predictions must be as precise as possible because a testing user may use them soon in the real world.

3.2.7.2 Software Interfaces of the System

The system uses the following software: IDLE .Due to its use case, the overall system must be adaptable, simple to use, and have a high level of accuracy (recall metric).

3.2.8 Non-Functional Requirements

Indirectly related to the specifications that the system is expected to meet, these requirements definE criteria that can be used to judge or evaluate the system's performance rather than specific behavior.

Efficiency: The system must operate as efficiently as possible under all conditions

Availability: The system needs to be always accessible.

Security: The system must be solidly constructed and free from security risks

Uniformity: The system must operate consistently across all web browsers.

Speed: The programme needs to be responsive and have a short latency period.

Accuracy: The system must make a reliable forecast.

3.3 Summary

The specific requirements and limitations that must be taken into account when designing the application are thoroughly explained in this chapter. These include the hardware and software requirements as well as the functional and non-functional requirements for the application. This chapter contains a complete list of the requirements that have been mutually agreed upon by the client and the software developer. Additionally, it details the system's capabilities and restrictions.

CHAPTER-4
HIGH LEVEL DESIGN

CHAPTER 4
High Level Design

In the Design Phase of the Software Development Life Cycle (SDLC), the architecture of the system must be created. The organizational structure of the project is explored and the structure of the system is conceptualized. The application is divided into progressively smaller parts known as services. These services are modular and are expected to have strong cohesion and low coupling in order to lessen interdependence. The outcome of this phase is a precise, modular design plan that specifies the functions of each component and how those functions are conceptually implemented when the components are merged. A system overview and a description of the overall software architecture are provided by high level design.Design Consideration

The system and its architecture are designed during the Design Phase of the Software Development Life Cycle (SDLC). The system's structure is conceived of and illustrated, and the project organization is defined. The application is broken up into increasingly smaller services, or modules. Since these services are modular, it is anticipated that they will exhibit strong cohesion and low coupling to reduce interdependence. The output of this phase is a clear, modular design plan that describes the roles of each component and how, when combined, the components conceptually realize the SRS's needs. High Level Design serves as a system overview and a means of elaborating on the entire software architecture.

4.1.1 General Constraints

Here are some general restrictions that must be considered:

- To use the system, the user needs use high-quality images (pixels).
- The user needs to have the fundamental knowledge to operate the system correctly.
- For the system to produce accurate results, the user's data must be sufficient.

4.1.2 Development Methods

In this chapter, the design methodology is the main topic. The data flow model provides a suitable flow or notion about the design strategy used for the application's development. A data flow model depicts how information is transferred as it is processed. Data is placed in a standard file to indicate the functional processing because the overall size is relatively little. The data flow model provides adequate documentation for each sub-model, which helps us understand how the data is related to each one.

4.2 Architectural Strategies

This Section provides the system's overall structure, as well as its submodules and comprehensive design. Additionally, it offers important system design understanding that may be used to develop applications without flaws. The application must meet each of the outlined standards.

4.2.1 Programming Language

The programming language used for this project is Python, which is a versatile and widely adopted language in the field of machine learning and data science. Python offers a rich ecosystem of libraries and frameworks that are well-suited for developing and implementing machine learning models.For the creation of the augmented dataset and preprocessing of images, Python provides libraries such as OpenCV and scikit-image, which offer a range of image processing functionalities. These libraries enable the application of various augmentation techniques, including rotations, translations, flips, and color transformations.

TensorFlow, a well-liked machine learning package for Python, offers a stable framework for convolutional neural networks (CNN) and other deep learning models. CNNs are suitable for underwater mine detection since they are very good at image classification jobs. The CNN model can also be built and trained using other libraries like Keras and PyTorch. Python's simplicity, readability, and extensive documentation make it an ideal choice for this project. Its vast community support ensures access to numerous resources, tutorials, and pre-built implementations, facilitating the development and evaluation of the CNN model and the comparison of results with the benchmarked dataset.In summary, Python's versatility, rich libraries, and strong community support make it the programming language of choice for this project, enabling efficient development and implementation of machine learning models for underwater mine detection.

4.2.2 Train-Test Split

The dataset used for our study was created by manually compiling photographs from a variety of online sources. Although the dataset was initially somewhat limited, it has been expanded by the use of appropriate augmentation techniques, which also improve image prediction.80% of the time is spent on training and 20% on testing in this split.

4.2.3 Data Storage Management

The images and the models are the only data that have been used for the purpose of this study.The images which are provided as the input by the user are stored in the user's local directory.Images are also provided as the input for preprocessing techniques as well as the augmentation process. The output images are also stored in the local directory. The CNN architectures are trained, the trained models are saved onto the same local folder and then loaded into the prediction m. These loaded models are used to predict the input image uploaded by the user to the test folder.

4.3 SystemArchitecture

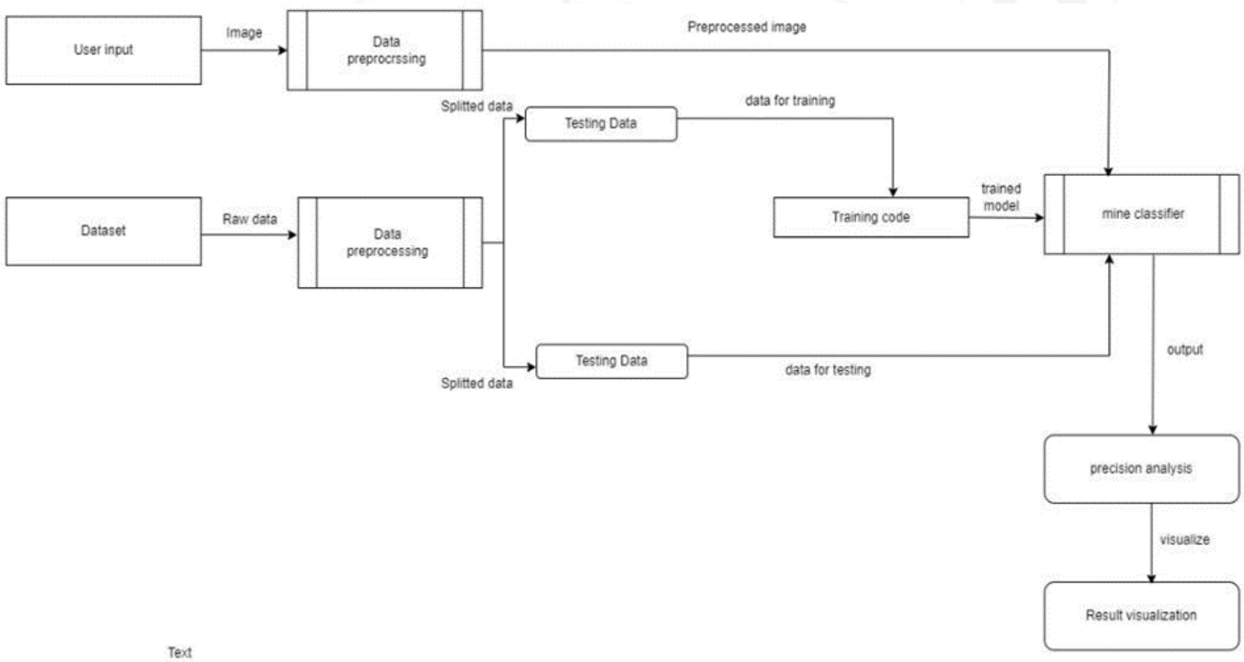


Fig 4.3.0. System Architecture

This model is used to create an augmented dataset for underwater mine detection using augmentation techniques. A Convolutional Neural Network model is trained on the dataset and compared with a benchmarked dataset to evaluate its performance, enhancing accuracy of underwater mine detection.

4.4 Data Flow Diagrams

The complete system's data flow is graphically represented in the Data Flow Diagram (DFD). The main focus of data flow models is on the data flow between distinct submodules through various processing processes. Data flow diagrams are typically composed of four basic components: processes, data flows, external entities, and internal entities. The user can better comprehend how the entire system works by seeing the flow. DFDs provide end users with information on how data is processed and the effects it will have on the overall system when some data is supplied as system input and eventually the system's output is produced.

4.4.1 Data Flow Diagram - Level 0

The overall operation of the system is described in the Level 0 DFD. The user, the system, and the inputs and outputs between the user and the system are all represented. The system's Level 0 DFD is displayed below.

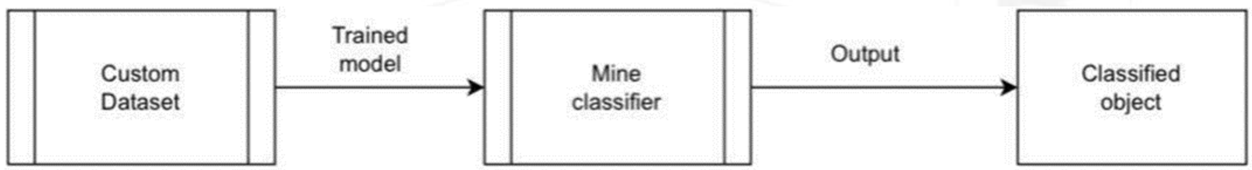


Fig 4.4 Level 0 DFD for Naval mine Detection

4.4.2 Data Flow Diagram - Level 1

The system is precisely broken down into submodules in the Level 1 Data Flow Diagram. The system's Level 1 DFD is displayed below.

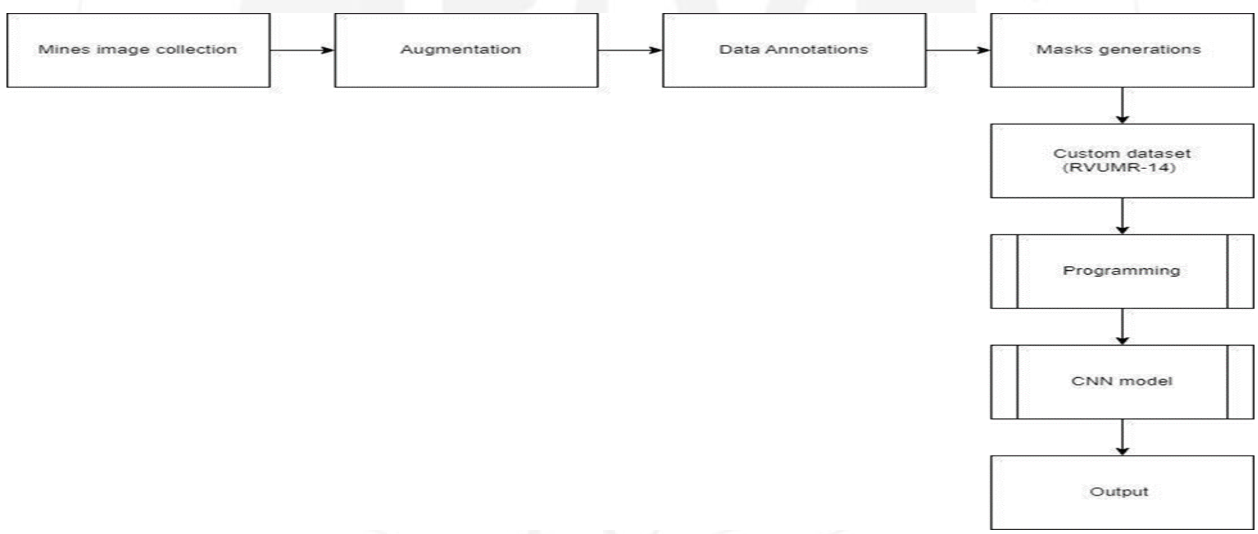


Fig 4.5 Level 1 DFD for Underwater Naval Mine Detection System

4.4.3 Data Flow Diagram - Level 2

In Level 2 DFD, the procedures from Level 1 DFD are expanded. As illustrated in the pictures below, Level 2 DFD provides a full description of the system. The figure below elaborates on the process of Image Preprocessing, specifically Denoising

Fig 4.6 Level 2 DFD for Underwater Naval Mine Detection System

The process of image preprocessing, in particular image data augmentation and annotation, is expanded upon in the figure below.

Fig 4.7 Level 2 DFD for Underwater Naval Mine Detection System

The technique of training CNN models and using these pre-trained models to identify sea mines in user- input photos is further explained in the graphic below

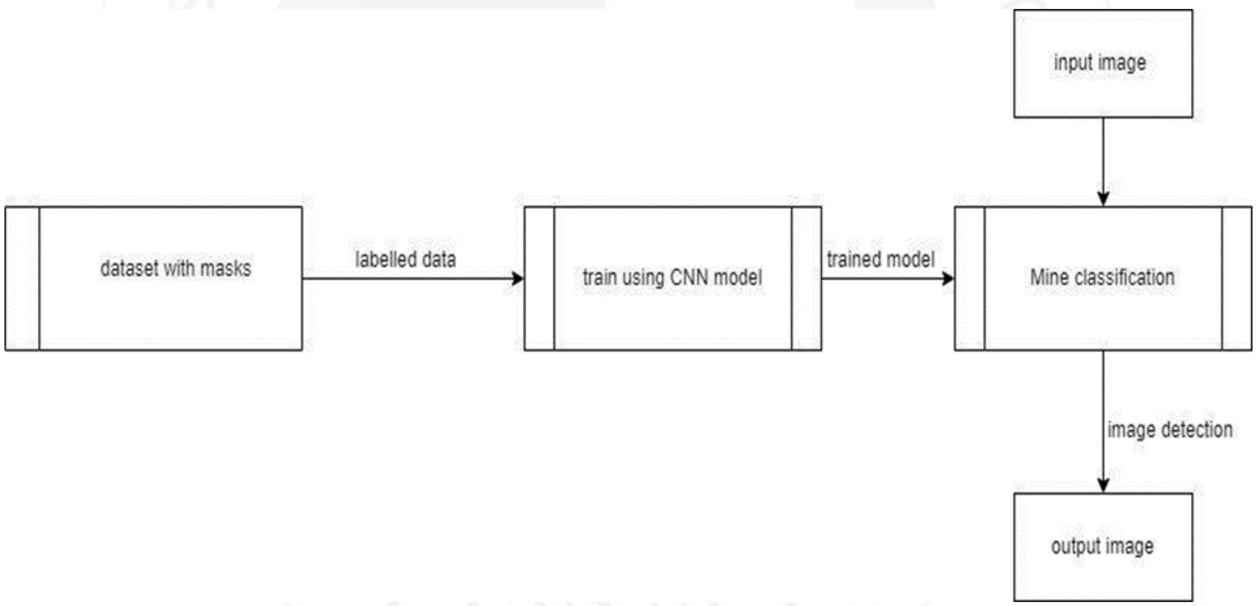


Fig 4.8 Level 2 DFD for Underwater Naval Mine Detection System

4.4 Summary

The above-drawn data models show how data enters and leaves the system as well as how the system processes it. This chapter's main objective was to outline the system's high-level design

CHAPTER-5
DETAILED DESIGN

CHAPTER 5

Detailed Design of Detection and classification.

Each module in the High-Level Design Plan is developed further in this phase, and its internal workings are thought out and described. The bottom level's detailed components and subcomponents. As a result, a complete design plan that includes additional but crucial information relevant to each module, such as input and output type, potential data structures, and algorithms employed, is produced. The proposed system's control flow between the modules is shown graphically in the Detailed Design using a Structure Chart, which also shows the modular hierarchy. You may read more about the modules in the following sections.

5.1 Structure Chart

The map of the structure chart shows the control flow inside the system among the modules and the modular hierarchy. It describes all the modules found, and how the modules communicate. It explains the submodules identified, too.

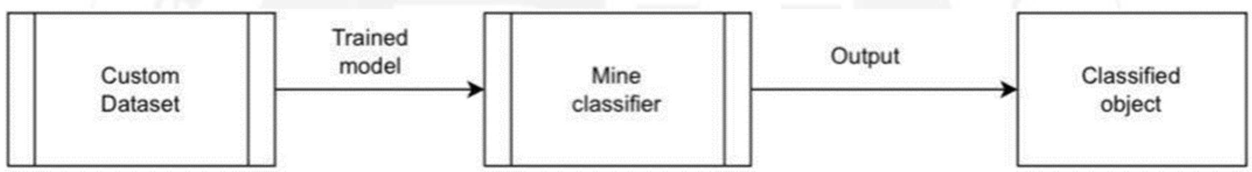


Fig.5.1.1 Structure chart

5.2 Functional Description of Modules

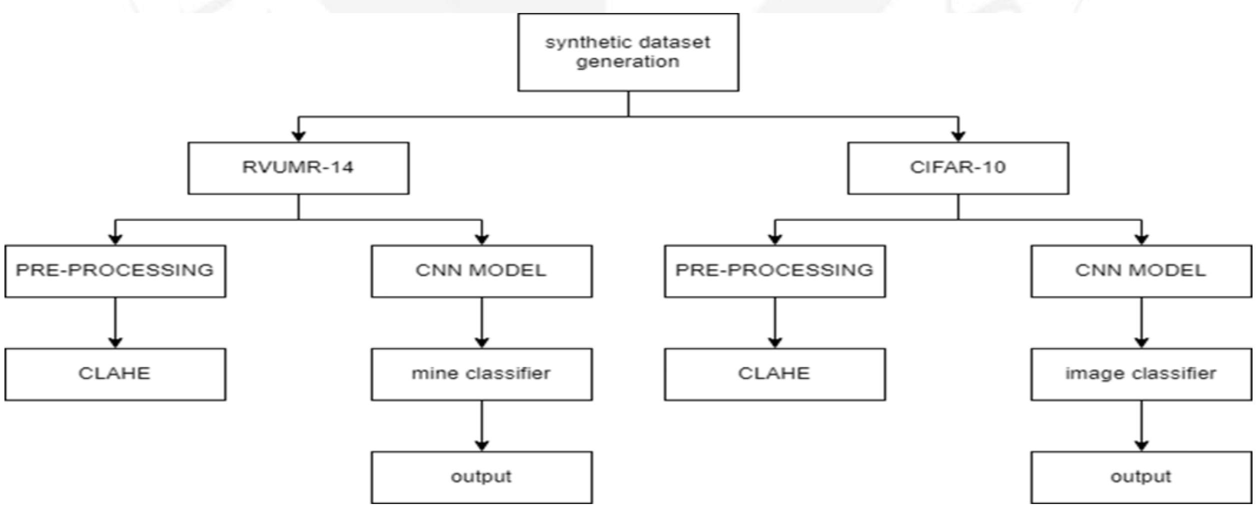


Fig 5.1 Structure Chart for Sea Mine Detection System

This section describes the internal functioning of each module. Also covered are software components and subcomponents.

5.2.1 Image Denoising Module

It uses Python script for loading and processing training images for a computer vision task. It uses the OpenCV library (`cv2`) to read and preprocess images, and the scikit-learn library (`sklearn`) for shuffling the data. The script defines several functions and a class to handle the loading and management of the dataset.The `load_train` function is responsible for reading and preprocessing the training images. It takes the `train_path` (the path to the directory containing the training images), `image_size` (the desired size of the images), and `classes` (a list of class names) as input. Inside the function, it iterates over the classes and reads all image files with the specified format (`*g`) in each class directory. For each image, it performs the following steps:

1. Reads the image using `cv2.imread`.
2. Resizes the image to the desired `image_size`.
3. Converts the image to `float32` data type and scales the pixel values between 0 and 1

4 .Appends the preprocessed image, corresponding label (one-hot encoded), image name, and class to the respective lists.

After processing all images, the lists are converted to NumPy arrays and returned as the output.The `DataSet` class is defined to represent a dataset. It takes the processed images, labels, image names, and classes as input during initialization. It provides several properties to access the dataset attributes such as images, labels, image names, number of examples, and the number of epochs done. Additionally, it implements the `next_batch` method to retrieve the next batch of examples from the dataset. It keeps track of the current index and updates it accordingly.The `read_train_sets` function is the main entry point for loading the training dataset. It takes the `train_path`,`image_size`, `classes`, and `validation_size` as input. It calls the `load_train` function to load and preprocess the images. Then, it shuffles the data using the `shuffle` function from scikit-learn. If the `validation_size` is a float, it converts it to the corresponding number of validation images based on the total number of images. It splits the shuffled data into validation and training sets using array slicing. Finally, it creates instances of the `DataSet` class for the training and validation sets and assigns them to the `train` and `valid` attributes of the `data_sets` object, which is returned as the output.Overall, this code provides a convenient way to load and preprocess training images for a computer vision task. It encapsulates the data in a `DataSet` object, which allows for easy access to images, labels, and other dataset attributes. The code also handles shuffling and splitting the data into training and validation sets.

The provided code implements a preprocessing technique called Contrast Limited Adaptive Histogram Equalization (CLAHE) on an image. This technique is commonly used in computer vision and image processing tasks for the following reasons:

1.Enhancing local contrast: CLAHE improves the visibility of details and enhances the contrast of an image. It redistributes the pixel intensities based on the local histogram within small regions (tiles) of the image. By limiting the contrast enhancement to a specified clip limit, CLAHE prevents the amplification of noise and avoids over-enhancement of the entire image.

2.Adaptive to image content: Unlike traditional histogram equalization, CLAHE adapts the enhancement to the local characteristics of the image. By dividing the image into tiles and independently equalizing each tile's histogram, CLAHE ensures that the enhancement is tailored to the specific content of different regions. This adaptivity helps to preserve the details and maintain a more natural appearance in the resulting image

3.Preserving color information: The code performs CLAHE in the HSV color space. By separating the image into its hue (H), saturation (S), and value (V) channels, CLAHE can be applied solely to the value channel. This approach avoids color distortion that may occur when applying histogram equalization techniques directly to RGB images. The code converts the modified HSV image back to RGB after CLAHE preprocessing. Overall, the code's purpose is to enhance the local contrast and improve the visual quality of the image while preserving its color characteristics. This can be beneficial in various computer vision applications, such as object detection, image segmentation, and feature extraction, by providing a more informative and visually enhanced input for subsequent processing steps.

The provided code block demonstrates a preprocessing step known as Contrast Limited Adaptive Histogram Equalization (CLAHE) applied to an image. Here's a step-by-step explanation of what the code does:

1.`hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)`: The input image (`img`) is converted from the BGR color space to the HSV (Hue, Saturation, Value) color space using OpenCV's `cv2.cvtColor` function. This conversion allows us to perform the CLAHE operation on the Value channel while preserving the color information.

2.`h, s, v = hsv_img[:,:,0], hsv_img[:,:,1], hsv_img[:,:,2]`: The HSV image is split into its individual channels: H (Hue), S (Saturation), and V (Value). These channels contain information about the color and intensity properties of the image.

3.`clahe = cv2.createCLAHE(clipLimit=2.5, tileGridSize=(20,20))`: A CLAHE object is created using OpenCV's

`cv2.createCLAHE` function. CLAHE is a variant of Histogram Equalization that operates on small regions (tiles) of an image, ensuring that the contrast enhancement is limited and adaptive to the local characteristics of the image.

4.`v = clahe.apply(v)`: CLAHE is applied to the V (Value) channel of the image. The CLAHE algorithm redistributes the pixel intensities within each tile based on the local histogram to improve the contrast while constraining the enhancement within a specified clip limit.

5.`hsv_img = np.dstack((h, s, v))`: The modified H, S, and V channels are stacked back together using `np.dstack` to form the modified HSV image. Now the V channel contains the contrast-enhanced version of the original image.

6.`image = cv2.cvtColor(hsv_img, cv2.COLOR_HSV2RGB)`: The modified HSV image is converted back to the RGB color space using `cv2.cvtColor`. This step is necessary to obtain the final preprocessed image in RGB format.

Overall, the code applies CLAHE to the V channel of the HSV image, enhancing the local contrast of the image while preserving its color information. The resulting preprocessed image, stored in the `image` variable, can potentially improve the visibility of image details and assist in subsequent computer vision tas

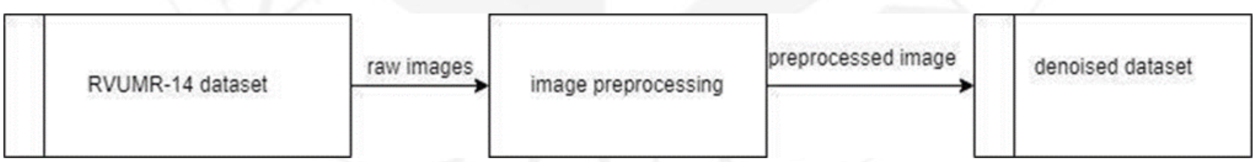
Fig 5.2.1: image

Fig 5.2.2: image with pre-processing(CLAHE).

Input: an raw image

Output: image with enhanced features

flowchart:



5.2.2 Data Augmentation Module

AUGMENTATION CODE:

```
from keras.preprocessing.image import ImageDataGenerator #generate images using keras
#from keras.preprocessing.image import img_to_array
```

```
from tensorflow.keras.utils import img_to_array from tensorflow.keras.utils import load_img
from tensorflow.keras.models import Sequential from tensorflow.keras import layers
import numpy as np import tensorflow as tf import cv2
import numpy as np from PIL import Image
from imgaug import augmenters as iaa import os
IMAGE_PATH = r"/content/drive/MyDrive/MAJOR PROJECT 89/MINES
ORIGINAL/AR0.jpg" OUTPUT_DIRECTORY = r"/content/drive/MyDrive/MAJOR
PROJECT 89/MINE 1/m3"
image = load_img(IMAGE_PATH) image = img_to_array(image)
image = np.expand_dims(image, axis=0) image1 = cv2.imread(IMAGE_PATH)
#       def to_grayscale_then_rgb(image):
image = tf.image.rgb_to_grayscale(image) image = tf.image.grayscale_to_rgb(image) return
image
datagen_compound= ImageDataGenerator( featurewise_center=True,
featurewise_std_normalization=True, rotation_range=20, width_shift_range=0.2,
height_shift_range=0.2, horizontal_flip=True, validation_split=0.2, brightness_range=[0.2,5.0]
width_shift_range = 0.2,
height_shift_range = 0.2, rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip =
True, fill_mode = 'nearest', data_format='channels_last', brightness_range=[0.1,0.9]
)
datagen_compound3=ImageDataGenerator( rescale=1. / 256,
rotation_range=75, width_shift_range=0.3, height_shift_range=0.1, fill_mode='constant',
cval=0, zoom_range=[.9, 1.25],
)

datagen_compound4=ImageDataGenerator( rotation_range =50,
width_shift_range = 0.2,
height_shift_range = 0.2, rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip =
True, fill_mode = 'nearest', data_format='channels_last',
preprocessing_function=to_grayscale_then_rgb
)

#
seq1 = iaa.Sequential([ iaa.Fliplr(0.5), iaa.Crop(percent=(0, 0.1)), iaa.Sometimes(
0.5,
iaa.GaussianBlur(sigma=(0, 0.5))
```

```
),
iaa.LinearContrast((0.75, 1.5)),
iaa.AdditiveGaussianNoise(loc=0, scale=(0.0, 0.05*255), per_channel=0.5), iaa.Multiply((0.8, 1.2), per_channel=0.2),
iaa.Affine(
scale={"x": (0.8, 1.2), "y": (0.8, 1.2)},
translate_percent={"x": (-0.2, 0.2), "y": (-0.2, 0.2)},
rotate=(-25, 25),
shear=(-8, 8)
)
], random_order=True) imglist1 = []
img1 = np.asarray( image1 ) seq2 = iaa.Sequential([iaa.Fliplr(0.2), iaa.Crop(percent=(0, 0.2)), iaa.Sometimes(
0.6,
iaa.GaussianBlur(sigma=(0, 0.2))
),
iaa.LinearContrast((0.35, 2.5)),
iaa.AdditiveGaussianNoise(loc=0, scale=(0.0, 0.06*255), per_channel=0.5), iaa.Multiply((0.8, 1.2), per_channel=0.2),
iaa.Affine(
#         scale={"x": (0.8, 1.2), "y": (0.8, 1.2)},
#         translate_percent={"x": (-0.2, 0.2), "y": (-0.2, 0.2)},
rotate=(-45, 45),
shear=(-8, 8)
)
], random_order=True)

imglist2 = []
img2 = np.asarray( image1 )

seq3 = iaa.Sequential([ iaa.Fliplr(0.2), iaa.Crop(percent=(0, 0.2)), iaa.Sometimes(
0.6,
iaa.GaussianBlur(sigma=(0, 0.2))
),
iaa.GammaContrast((0.5, 2.0)),
```

```
iaa.AdditiveGaussianNoise(loc=0, scale=(0.0, 0.06*255), per_channel=0.5), iaa.Multiply((0.8,
1.2), per_channel=0.2),
iaa.Affine(
#       scale={"x": (0.8, 1.2), "y": (0.8, 1.2)},
#       translate_percent={"x": (-0.2, 0.2), "y": (-0.2, 0.2)},
rotate=(-195, 195),
#       shear=(-8, 8)
)
], random_order=True)

imglist3 = []
img3 = np.asarray( image1 )
#

PREFIX = 'img_'
imGen1 = datagen_compound.flow(image, batch_size=1, save_to_dir =
OUTPUT_DIRECTORY, save_prefix=PREFIX, save_format='jpg')
imGen2 = datagen_compound1.flow(image, batch_size=1, save_to_dir =
OUTPUT_DIRECTORY, save_prefix=PREFIX, save_format='jpg')
imGen3 = datagen_compound2.flow(image, batch_size=1, save_to_dir =
OUTPUT_DIRECTORY, save_prefix=PREFIX, save_format='jpg')
imGen4 = datagen_compound3.flow(image, batch_size=1, save_to_dir =
OUTPUT_DIRECTORY, save_prefix=PREFIX, save_format='jpg')
imGen5 = datagen_compound4.flow(image, batch_size=1, save_to_dir =
OUTPUT_DIRECTORY, save_prefix=PREFIX, save_format='jpg')
#        for i in range(20):
                images_aug1 = seq1(image=img1)
```

Creating numerous versions of an image in order to expand a dataset is known as "image augmentation" in the field of computer vision. Images can be enhanced to produce fresh data that will aid machine learning models in understanding the underlying patterns in the data.Images can be improved in a variety of ways, including by rotating, cropping, flipping, altering the brightness and contrast, and adding noise. Each of these methods can create numerous iterations of an image, each with its own distinct features. For instance, rotating an image can result in a new image with a similar appearance to the original but with a different rotation angle. When training models to recognise particular objects or features, cropping an image can result in new images that only contain a piece of the original image.When working with tiny datasets, enhancing photos can be very helpful because it can enhance the number of images available for training without requiring additional data gathering. The risk of overfitting, which happens when a model becomes overly specialized to the training data and performs badly on fresh, unknown data, can also be decreased by producing numerous versions of each image.To sum up, image augmentation is an effective method for producing numerous versions of an image that may be used to expand a dataset and enhance the effectiveness of machine learning models. We may generate fresh data by experimenting with various augmentation methods, which will enable our models to learn more effectively and produce more precise predictions.This code imports a number of modules and functions needed to enhance image data. By applying several transformations to the current photos, such as rotation, scaling, flipping, shearing, etc., image data augmentation creates new images.

This method is frequently used to expand the training data set, which enhances the effectiveness of the machine learning model.The first few lines of the code import the required modules such as ImageDataGenerator from keras.preprocessing.image, img_to_array from tensorflow.keras.utils, and load_img from tensorflow.keras.utils. These modules are used for image manipulation and conversion.Next, the Sequential model from tensorflow.keras.models and the layers module from tensorflow.keras are imported. These are used for building the machine learning model.The numpy module is imported with the alias np, and the cv2 and PIL modules are also imported for image processing.Then, the to_grayscale_then_rgb function is defined. This function takes an image and converts it to grayscale using the tf.image.rgb_to_grayscale function, and then converts it back to RGB format using the tf.image.grayscale_to_rgb function. This is done to add grayscale images to the data set.The ImageDataGenerator class is used to generate new images by applying different transformations to the existing images. In this code, four instances of the ImageDataGenerator class are created: datagen_compound, datagen_compound1, datagen_compound2, and datagen_compound3.

Each of these instances is initialized with different transformation parameters such as rotation range, width shift range, height shift range, horizontal flip, brightness range, etc.The seq1, seq2, and seq3 variables are initialized with Sequential instances from the imgaug module. These variables are used to apply additional image augmentations to the existing images. Each of these Sequential instances is defined with a set of image augmentation techniques such as flipping, cropping, Gaussian blur, linear contrast, additive Gaussian noise, affine transformation, gamma contrast, etc.The IMAGE_PATH variable is used to specify the path of the original image that needs to be augmented. The OUTPUT_DIRECTORY variable is used to specify the directory where the augmented images will be saved.The load_img function is used to load the original image from the specified path, and the img_to_array function is used to convert the loaded image to a numpy array. The np.expand_dims function is then used to add an extra dimension to the numpy array.Finally, the cv2.imread function is used to read the original image in OpenCV format and save it as image1. This is done to apply some of the augmentations later in the code.Overall, this code is used to perform image data augmentation on an original image. It applies various transformations to the original image and generates new images. These new images are saved in the specified output directory. The augmented images can then be used to improve the performance of a machine learning model.generates augmented images using both Keras and imgaug libraries.

In Keras, ImageDataGenerator is used to create an image data generator with several parameters for image augmentation, such as rotation range, width shift range, height shift range, horizontal and vertical flip, brightness range, and others. These parameters are used to create a generator that can randomly apply these operations to images.In imgaug, Sequential and Sometimes are used to create sequences of augmentations with random order and probabilities. Several augmentations are used such as flipping, cropping, Gaussian blur, linear contrast, additive Gaussian noise, multiply, affine transformations, gamma contrast, among others.Both Keras and imgaug libraries allow creating a variety of augmented images by applying different combinations of operations to the original image. The generated images can be useful to improve the performance of machine learning models by increasing the size of the training dataset and preventing overfitting.It generates data augmentation for image data using Keras and imgaug libraries. The script imports the necessary modules for data augmentation using Keras, such as ImageDataGenerator, Sequential, layers, and other libraries like cv2, numpy, and PIL.The script loads an image from a specified path, converts it to an array, and applies various data.

Augmentation techniques. It defines multiple ImageDataGenerator objects, each with a specific set of data augmentation parameters. The data augmentation techniques include rotation, flipping, shearing, zooming, adding noise, and changing brightness and contrast.The script also uses the imgaug library to define three additional augmentation pipelines. Each pipeline applies a set of transformations that include flipping, cropping, blurring, adding noise, and contrast adjustments. The code creates a list of augmented images for each pipeline.Overall, this script allows for generating a large number of augmented images from a single input image, which can be useful for improving the performance of machine learning models that use image data.

Input : one image from different classes of mine

Output: generates a dataset with augmentation technique

flowchart:

5.2.3 Data Labeling Module

This module assigns the labels to different types of Sea Mine to each image manually in the image collection.The platform used for this purpose was Apeer.com.Apeer.com is an online platform that provides a collaborative environment for researchers to create, share, and execute machine learning workflows. In the context of generating masks for the above dataset, apeer.com can be utilized as follows:

1.Workflow Creation: Researchers can create a workflow on apeer.com specifically designed for generating masks from underwater mine images. The workflow can include a series of steps and modules to process the images and generate corresponding masks.

2.Image Processing Modules: Researchers can incorporate various image processing modules available on apeer.com into the workflow. These modules can include operations such as thresholding, edge detection, or segmentation algorithms specifically tailored for generating masks.

3.Collaboration and Sharing: The apeer.com platform allows researchers to collaborate with team members or other contributors by sharing the created workflow. This enables teamwork and facilitates feedback and improvements in the mask generation process.

4.Workflow Execution: Once the workflow is defined, researchers can execute it on apeer.com. The platform provides the necessary computational resources to perform the image processing operations and generate masks for the underwater mine images.

5.Iterative Refinement: Researchers can iterate and refine the workflow as needed based on the results and feedback obtained from the mask generation process. This iterative approach allows for continuous improvement and optimization of the mask generation process.

By utilizing apeer.com, researchers can leverage the platform's resources, modules, and collaborative features to streamline the process of generating masks for the above dataset. The platform's flexibility and accessibility enable efficient experimentation, workflow execution, and collaboration, ultimately enhancing the quality and accuracy of the generated masks.This module's goal is to label the dataset of images.The Augmented dataset of photographs will be the input given to this module.Output: The photos that have been labeled and categorised will be this module's output.Functionality: This module's functionality is to label and classify the image input data into different classes of mines.

Input: raw images

Output: masks with images

flowchart:

5.2.4 Sea Mine Detection Model Training and Testing Module

Input: The labeled dataset of 14 different classes of sea mines will be the input given to this module.

Output: This module's output will be trained and tested sea mine detection models, which categorize mines into 14 types.

Functionality: This module's functionality entails using the tagged image dataset to train, test, and provide trained models for use in other modules.

It uses TensorFlow implementation of a convolutional neural network (CNN) for image classification. It trains the CNN using a dataset of images and displays the training progress, including accuracy and loss, for each epoch. It also saves the trained model for future use. Finally, it plots the trend of accuracy and loss over the epochs and calculates the execution time of the code.The code begins by importing the required libraries and disabling eager execution in TensorFlow. It then sets up various parameters such as the number of iterations, batch size, image size, and number of channels.

Next, it prepares the input data by reading and processing the images from a specified training directory. It divides the data into training and validation sets and prints some information about the dataset.The network architecture is defined using convolutional, flatten, and fully-connected layers. The weights and biases are created as TensorFlow variables, and the layers are connected using appropriate activation functions. The output of the network is passed through a softmax function to obtain class probabilities.The code then defines the loss function (softmax cross-entropy) and the optimizer (Adam optimizer) for training the network. It also calculates the accuracy of the predictions. During the training loop, batches of images and labels are fed into the network, and the optimizer is run to minimize the loss. The progress is displayed for each epoch, and the trained model is saved.After training, the code plots the accuracy and loss trends over the epochs using matplotlib. Finally, the execution time of the code is calculated and displayed.In summary, this code trains a CNN for image classification, monitors the training progress, saves the trained model, and visualizes the accuracy and loss trends.In the above code, the calculation of epochs is based on the number of iterations performed during training. The variable `total_iterations` keeps track of the total number of iterations completed so far. Within the training loop, for each iteration `i`, the following check is performed:

```python
if i % int(data.train.num_examples/batch_size) == 0:     ---------(a)
```

This condition checks if the current iteration is a multiple of the number of iterations required to process the entire training dataset (`data.train.num_examples/batch_size`). If this condition is true, it indicates that one epoch has completed.The calculation of `total_epochs` within the condition is as follows:

```python
total_epochs = int(num_iteration/int(data.train.num_examples/batch_size)) + 1  --------(b)
```

Here, `num_iteration` refers to the total number of iterations to be performed during training (specified by `final_iter` in the code). The division `num_iteration/int(data.train.num_examples/batch_size)` gives the total number of epochs minus one. Adding 1 accounts for the current epoch being completed.For example, if `num_iteration` is set to 1000 and the training dataset has 10000 examples with a batch size of 21, then `int(data.train.num_examples/batch_size)` is 476 (approximately). Thus, `total_epochs` would be `1000/476 + 1 = 3` (approximately).The `show_progress` function is called when one epoch completes, and it takes the current epoch number, training and validation data feed dictionaries, validation loss, and total number of epochs as input parameters. It prints the training and validation accuracies and the validation loss for that epoch.In the above model, the Convolutional Neural Network (CNN) plays a crucial role in the process of image classification. CNNs are a specialized type of neural network architecture designed specifically for processing grid-like data, such as images. They are particularly effective in capturing local patterns and spatial relationships in images.

The role of the CNN in the above model can be broken down into several key functions:

Feature Extraction: The initial layers of the CNN are responsible for extracting relevant features from the input images. These layers consist of convolutional filters that scan the input image using a sliding window approach. Each filter applies a convolution operation to a small receptive field, capturing local patterns and detecting features such as edges, corners, and textures. By using multiple filters, the CNN can learn a diverse set of features at different scales and orientations.

Non-linear Activation: After the convolutional operations, activation functions are applied to introduce non-linearity into the network. In the above model, Rectified Linear Unit (ReLU) activation functions are used, which allow the CNN to model complex and non-linear relationships between features. ReLU activation sets negative values to zero and keeps positive values unchanged, helping in removing linearity and introducing non-linear transformations.

Pooling: Following the activation layers, pooling operations are applied to reduce the spatial dimensionality of the feature maps. Max-pooling is commonly used, which partitions the input into small non-overlapping regions and outputs the maximum value within each region. Pooling helps in downscaling the feature maps, reducing computational complexity, and improving the model's robustness to small spatial variations.

Flattening: The output feature maps from the convolutional and pooling layers are flattened into a one-dimensional vector. This process rearranges the spatial information into a linear representation, allowing the features to be fed into the subsequent fully-connected layers.

Classification: The flattened features are then passed through one or more fully-connected layers. These layers perform a linear transformation of the input followed by an activation function (ReLU in the above model). The fully- connected layers learn to map the extracted features to the corresponding classes in the dataset. The final layer usually employs a softmax activation function to produce class probabilities.

By utilizing multiple convolutional, pooling, and fully-connected layers, the CNN is capable of automatically learning and extracting hierarchical representations of the input images. The initial layers learn low-level features, such as edges and textures, while deeper layers learn more abstract and high-level features. This hierarchical feature extraction enables the CNN to capture intricate patterns and discriminative features that are crucial for accurate image classification.Overall, the CNN's role in the above model is to extract and transform the input images into meaningful feature representations, which are then used for classification and prediction. Its ability to learn and exploit spatial relationships in images makes it a powerful tool for tasks like image recognition, object detection, and other visual perception tasks. #role of optimizer and learning rate.In the above model, the optimizer and learning rate play important roles in training the convolutional neural network (CNN) model. They contribute to the process of updating the model's weights and biases based on the computed loss during training.The optimizer is responsible for minimizing the loss function of the model. In this code, the Adam optimizer is used, which is a popular optimization algorithm for deep learning models. It combines the benefits of both AdaGrad and RMSProp optimizers by adapting the learning rate for each parameter based on its first and second-order moments.

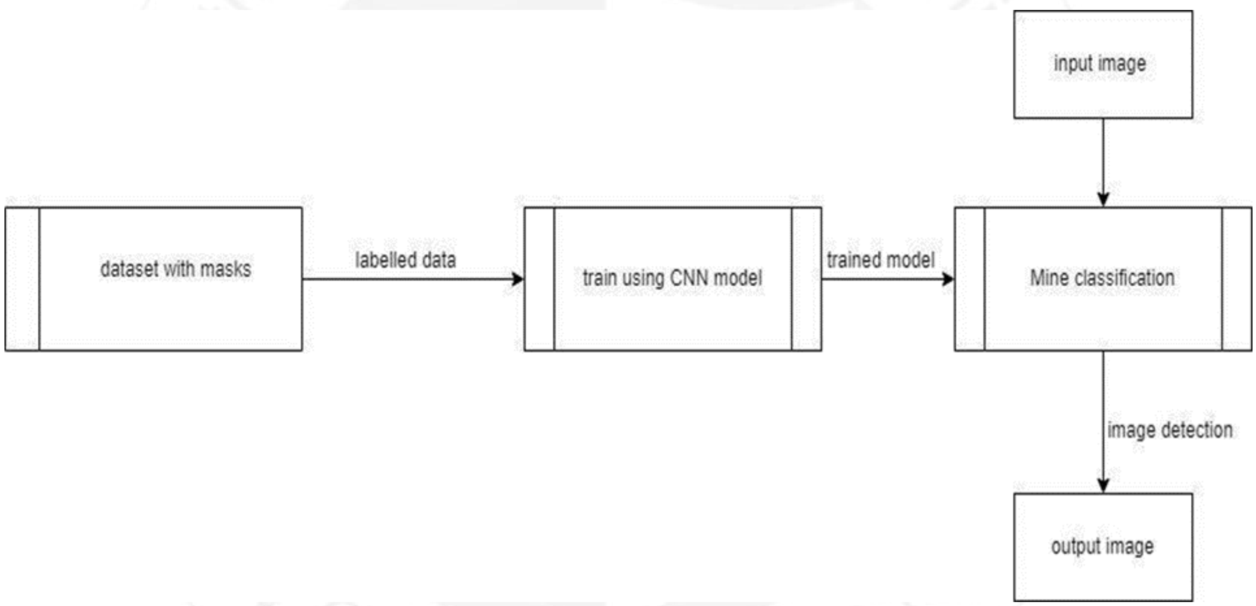
The optimizer, specifically `tf.train.AdamOptimizer`, is instantiated with a specified learning rate (`learning_rate=1e- 4`). The learning rate determines the step size at which the optimizer adjusts the model's parameters during each iteration. A larger learning rate allows for more significant updates but may lead to overshooting the optimal solution, while a smaller learning rate may converge slowly. The choice of an appropriate learning rate is crucial in finding a balance between convergence speed and optimization stability.During training, the optimizer's `minimize()` method is called, passing the loss function (`cost`) as an argument. This step calculates the gradients of the loss with respect to the model's trainable parameters (weights and biases) and updates them accordingly using the optimization algorithm.By iteratively updating the model's parameters based on the gradients, the optimizer guides the model towards minimizing the loss function and improving its ability to make accurate predictions on the training data.The learning rate, in combination with the optimizer, determines the speed and quality of the model's convergence during training. If the learning rate is set too high, the model may fail to converge, resulting in unstable training and potentially overshooting the optimal solution. On the other hand, setting the learning rate too low may cause slow convergence and longer training times.Finding an optimal learning rate often involves experimentation and tuning. If the learning rate is not properly chosen, the model's performance may suffer. Techniques such as learning rate schedules or adaptive learning rate algorithms can be employed to adjust the learning rate during training to strike a balance between convergence speed and stability.

In summary, the optimizer and learning rate work together to update the model's parameters and guide the training process. The optimizer minimizes the loss function by adjusting the weights and biases, while the learning rate controls the step size of these adjustments. Proper selection and tuning of the optimizer and learning rate are essential for achieving good training performance and convergence of the CNN model.

Input: dataset (RVUMR-14 and CIFAR-10)

Output: precision metrics

Flowchart:



Choosing the best values for the model:

For a dataset with 2100 images, the choice of the best batch size and final number of iterations (`final_iter`) can vary depending on factors such as the model complexity, computational resources, and the trade-off between training time and model performance. Given the smaller dataset size, here are some general guidelines to consider:

Batch Size:

1.Training Stability: With a smaller dataset, you can experiment with larger batch sizes without encountering memory limitations. Larger batch sizes may provide a more stable gradient estimation.

2.Generalization: Smaller batch sizes introduce more randomness into the weight updates, which can act as a regularizer and potentially improve generalization. However, very small batch sizes may result in noisy gradients and slower convergence.

Considering the dataset size of 2100 images, a batch size in the range of 16 to 64 could be a good starting point. You can begin with a moderate batch size, such as 32, and observe the training dynamics. If the model converges smoothly and you have sufficient computational resources, you can experiment with larger batch sizes to see if they offer any benefits.

Final Iterations:

1.Convergence: Monitor the loss and accuracy during training. The training process should be run until the loss stabilizes or reaches a satisfactory level, and the accuracy plateaus or improves marginally. You can observe the training progress and decide on the number of iterations accordingly.

2.Early Stopping: If the model's performance on a validation set stagnates or deteriorates, it might indicate that further training is not beneficial. Apply early stopping to halt training before reaching the maximum number of iterations.

Considering the smaller dataset size, a lower number of iterations, such as 500 to 1000, could be sufficient to achieve good results. Start with a conservative value and assess the model's performance. If you observe that the model is not converging or the performance improvements are minimal, you can extend the number of iterations by increasing

`final_iter`. It is important to note that these suggestions are not definitive and should serve as starting points for experimentation. The best batch size and final number of iterations can vary depending on the specific dataset and model. It is recommended to try different configurations, monitor the training process, and assess the model's performance to determine the optimal values for your particular case.

5.2.5 Sea Mine Detection Module

It imports necessary libraries such as `sklearn.metrics`, `tensorflow`, `numpy`, `os`, `cv2`, `time`, and `matplotlib.pyplot`. It disables eager execution in TensorFlow and starts a timer.The code then checks if the specified directories for training and testing images exist. If not, it raises an exception. It then iterates through the files in the testing directory.For each file, it reads the image using OpenCV, resizes it to a desired size, and preprocesses it. The preprocessed image is fed into a trained TensorFlow model for prediction. The model is restored, and the input and output tensors are obtained from the graph. The image is passed through the model, and the predicted class probabilities are obtained.The code finds the class with the highest predicted probability and determines the corresponding label. It checks if the maximum confidence output is significantly larger than the other outputs.

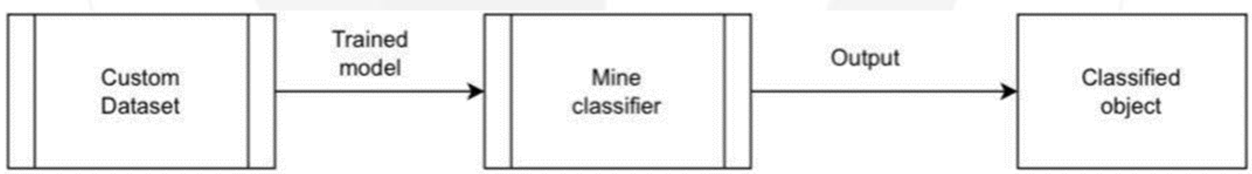
If so, it prints the predicted class; otherwise, it prints a warning.If the file does not exist, it prints a message indicating that the file is missing.After the execution of the prediction loop, the code calculates the total execution time and prints it.Next, it calculates the F1 score and mean squared error (MSE) using the predicted class probabilities and a true label array (`y_true`). The F1 score is a measure of the model's accuracy, and MSE quantifies the average squared difference between the predicted probabilities and true labels.The code then creates a confusion matrix using the true labels and rounded predicted probabilities. It visualizes the confusion matrix using matplotlib, displaying the matrix as a heatmap with labeled axes.

Overall, the code loads images, performs predictions using a TensorFlow model, evaluates model performance metrics (F1 score, MSE, recall, precision and confusion matrix), and visualizes the results.

Input: sample images

Output: predicted name and precision metrics

Flowchart:



5.3 Summary

This chapter describes the internal functioning of the five key modules with the essential dataflow via each of them. The structure chart offers a clear image of the control flow inside the system. The internal functioning of each individual is also explained by an independent.

CHAPTER 6 – IMPLEMENTATION OF DESIGN OF DETECTION AND EXTRACTION OF NAVAL MINE FEATURES SYSTEM

CHAPTER 6

Implementation of Detection and Classification of Underwater Mines

Because the design must be put into practise in order to solve the problem, the implementation phase is a crucial stage in the development of a project. In order to adhere to the SRS document criteria, low level/detailed designs will be converted into language-specific programmes during this step.This step includes the actual implementation of the concepts that were suggested in the design and analysis phase. The approaches and procedures for implementing software must promote reuse, make maintenance simple, and be widely understood. It is crucial to keep up the coding's discipline and pace.

6.1 Programming Language Selection

Python 3 is chosen to implement the Machine Learning Model development pipeline and to develop the Backend of the Web Based Sea Mine Detection System . The main reasons for using Python 3 are listed below.

- Python is a free and open-source programming language.
- The Language is simple to use and consistent.
- Python has a vast set of libraries specifically and extensively developed for Machine Learning and associated disciplines. Popular examples include Keras, Tensorflow, Pandas, Scikit ,OpenCV,etc.
- Python's extensive set web application frameworks can be extremely useful and timesaving, such as Flask, which has very less to no dependencies to external libraries,therefore light, flexible and easy to learn and implement.
- Platform Independent.

6.2 Platform Selection

An "idle" platform, in the context of programming, refers to a lightweight integrated development environment (IDE) or code editor that allows for writing and executing code without extensive features or complexities. It provides basic functionality such as code syntax highlighting, code execution, and debugging tools in a simple user interface.Idle platforms are commonly used for quick prototyping, testing code snippets, or running simple programs. They offer a straightforward and minimalistic environment for developers to write and execute code without the need for advanced features or complex development setups. Examples of idle platforms include IDLE for Python, Replit, or Jupyter Notebook in "playground" mode.However, idle platforms may have limitations in terms of scalability, advanced debugging capabilities, collaboration features, and integration with larger projects or complex workflows. Therefore, for more extensive projects, it may be necessary to switch to more robust development environments or platforms that offer comprehensive tools and features to support the project's requirements.

6.3 Code Conventions

The coding standards used throughout the project are discussed in this section. It contains the software libraries or APIs required to finish the project. Because huge projects should be coded in a globally recognised way, proper coding standards should be followed. This makes it much simpler for anyone to grasp any section of the code. Code conventions are significant becausethey promote readability and maintainability in software programmes, making it easier for programmers to grasp the code.

6.3.1 Naming Conventions

Naming conventions help to comprehend and easily comprehend the programs. Packs, scripts,graphs and classes must have clear and precise names in order to understand their contents quickly. The project is developed in Python and is called as follows:

Functions: should be a verb. The function name is a description of the function's function,separated by underscores.

Variables: Variable names must be short, meaningful and if multiple words are present,they are connected by an underscore. All letters are small.

6.3.2 File Organization

The file organization consists of three directory which consists of a collection of all the images ie. images containing both naval mines and its masks.These images are augmented and preprocessed and the output images obtained after adding to the same directory. Later, once the labeling process is done, the images are separated into two different directories namely sea mines and masks.Another directory with different classes of mines in respective class folders. Another consists of the benchmarked dataset.

6.3.3 Declarations

When coding in Python, standard declaration conventions are used. Modules are given common names to make it easier to understand the job of each specified item. Because of comments andto avoid ambiguity, several declarations per line are not permitted.

6.3.4 Comments

Comments are required in the construction of code since they assist coders in understanding the purpose of various functions and variables. Pair programming and code review benefit from comments as well. Each function has a commented part that explains the function in detail what that function accomplishes.

6.4 Difficulties Encountered during Implementation and Strategies Used toTackle Them

1. Lack of Sea Mine Image Data - The dataset has to be specially put together before being augmented to increase the number of samples because Seamine data is secret and difficult to obtain. To address this limitation, data augmentation techniques such as transformation, zooming, rotation, vertical and horizontal flipping, etc., are used.

2.Manual Labeling of the Images - all the images were manually labeled using the apeer platform.different colors were used for different classes of labels a shown the table_3.

6.5 Summary

This chapter describes the choice of programming language, the platform and describes their advantages. It also covers programming etiquette, such as function and variable name. It explains the structure of the project folders as well as the code comments. It also specifies the difficulties encountered during the implementation phase and how they are overcome

CHAPTER 7- SOFTWARE TESTING OF DETECTION AND EXTRACTION OF NAVAL MINE FEATURES SYSTEM

CHAPTER 7

SOFTWARE TESTING OF DETECTION AND EXTRACTION OF NAVAL MINE FEATURES SYSTEM

The software must be tested before, during, and after installation in order to ensure that it satisfies all requirements and provides the capabilities that the product's users expect. Verification and validation (V&V) involves evaluating the entire system as part of the process. Using test data to run a software application is a component of software testing. To ensure that the programme is operating as the client had anticipated, its outputs and operational behavior are reviewed. A dynamic method for successively assessing and verifying information is testing.

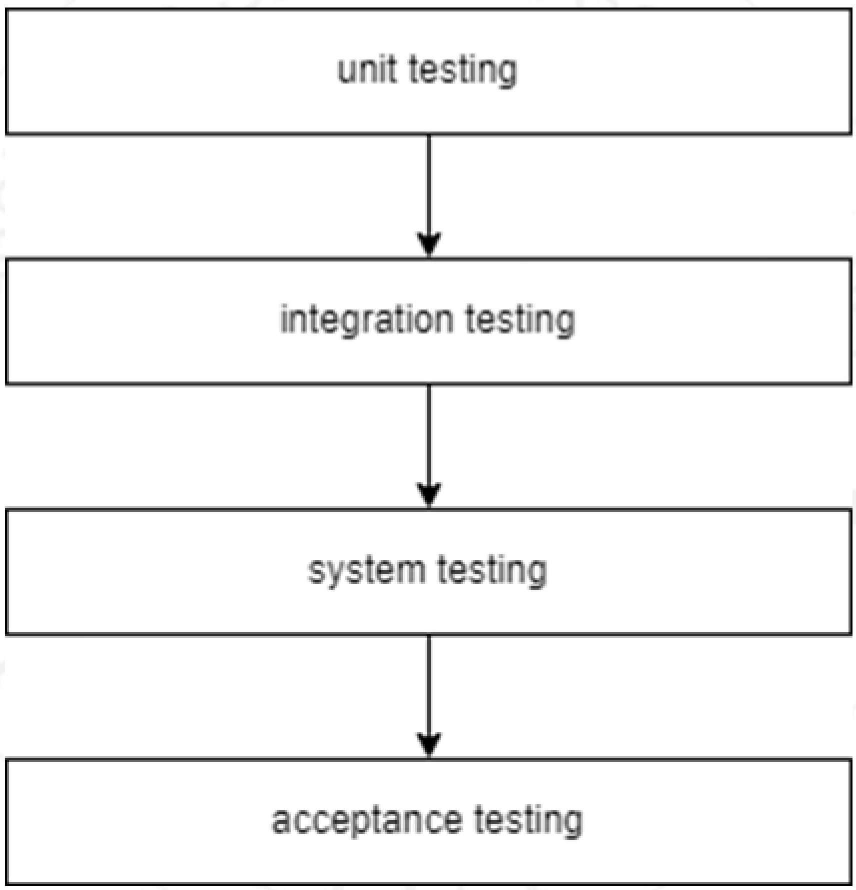


Figure 7.1 Phases of Testing

7.1 Test Environment

The currently developed software is tested in a testing environment, which comprises software and hardware. Should contain the following:

- Language:python

- OS: Windows/Linux/Mac
- Tool: idle, Python, OpenCV, Numpy, TensorFlow, Keras, Matplotlib
- Version : Python 3.9
- Dataset: RVUMR-14 Dataset

7.2 Unit Testing

Unit testing is a testing approach in which individual modules are checked by the developer to see if there are any flaws. It is concerned with the standalone modules' functional soundness. Following are the modules on which unit testing was performed along with test cases:

7.2.1 Unit Testing of RVUMR-14 dataset with and without CLAHE.

SL no of test case	1
Name of test case	RVUMR-14 (dataset.py&A_training.py)
Features being tested	Dataset pre-processing using CLAHE and Adam optimizer
Description	Checking its effect on our models training accuracy, validation accuracy, validation loss, trend of accuracy and trend of loss.
Sample Input	RVUMR-14(dataset pre-processing with or without CLAHE and ADAM OPTIMIZER)
Expected Output	Without CLAHE: Training Accuracy: 95.2%, Validation Accuracy: 85.7%, Validation Loss: 0.464
Actual output	Using CLAHE: Training Accuracy: 90.5%, Validation Accuracy: 95.2%, Validation Loss: 0.181
Remarks	The use of CLAHE gives high accuracy for our model and also the best value for learning rate =1e-4(Adam optimizer)

Table 7.1 Unit Testing of RVUMR-14

The first unit test (shown in Table 7.1) is for the the RVUMR-14 dataset.First CLAHE is used to pre-process dataset and load into the CNN model.Also, to smoothen the data, the adams optimizer is used checking its effect on our models training accuracy, validation accuracy and validation loss along with trend of accuracy and loss graphs.

7.2.2 Unit Testing of CIFAR-10 dataset with and without CLAHE.

SL no of test case	2
Name of test case	CIFAR-10 (dataset.py&training.py)
Features being tested	Dataset pre-processing using CLAHE and Adam optimizer
Description	Checking its effect on our models training accuracy, validation accuracy, validation loss, trend of accuracy and trend of loss.
Sample Input	CIFAR-10(dataset pre-processing with or without CLAHE and ADAM OPTIMIZER)
Expected Output	Using CLAHE :Training Accuracy: 81.0%, Validation Accuracy: 69.0%, Validation Loss: 0.885
Actual output	Without CLAHE :Training accuracy=76.2, validation accuracy=85.7, validation loss=0.754
Remarks	The use of CLAHE gives high training accuracy for our model and also the best value for learning rate =1e-4(Adam optimizer)

Table 7.2 Unit Testing of CIFAR-10

7.2.3 Integration testing of RVUMR-14 (prediction model)

SL no of test case	1
Name of test case	RVUMR-14 (prediction.py)
Features being tested	Image classification, F1_score, MSE, confusion matrix, recall and precision.
Description	Checking the accuracy and results of our model.
Sample Input	RVUMR-14(dataset pre-processing with CLAHE and ADAM OPTIMIZER)
Expected Output	MU2.png
Actual output	Predicted image: MU2 Execution Time: 0.20106959342956543 seconds f1 score for the model: 0.0 MSE for the model: 0.14285714285714285 Precision: [0.9230769230769231, 0.0] Recall: [0.9230769230769231, 0.0]
Remarks	The model predicted accurately with different values for metrics

Table 7.3 Integration testing of RVUMR-14 (prediction model)

7.2.1 Integration testing of CIFAR-10 (prediction model)

SL no of test case	2
Name of test case	CIFAR-10 (prediction.py)
Features being tested	Image classification, F1_score, MSE, confusion matrix, recall and precision.
Description	Checking the accuracy and results of our model.
Sample Input	CIFAR-10(dataset pre-processing with CLAHE and ADAM OPTIMIZER)
Expected Output	ship.png
Actual output	Predicted image: ship Execution Time: 0.2608664035797119 seconds f1 score for the model: 1.0 MSE for the model: 0.0 confusion matrix for the model: [[9 0] [0 1]] Precision: [1.0, 1.0] Recall: [1.0, 1.0]
Remarks	The model predicted accurately with different values for metrics

Table 7.3 Integration testing of CIFAR-10 (prediction model)

7.4 System Testing

System testing is the process of integrating all of the modules that have passed integration testing into a single framework. The device is assessed to ensure that all modules are properly linked to meet various user needs. This check aids in the entire bug removal process while also increasing device uniformity and confidence.During device testing, the gadget's appropriate operation is determined. This system testing evaluates the whole system, including all of the major components.

SL no of test case	3
Name of test case	RVUMR-14 vs CIFAR-10
Features being tested	Prediction accuracy, f1_score, MSE, training accuracy, validation accuracy, validation loss and confusion matrix
Description	Validating our dataset(RVUMR-14) with a benchmarked dataset(CIFAR-10)
Sample Input	1. RVUMR-14 2. CIFAR-10
Expected Output	CIFAR-10: training accuracy=81.0%, validation accuracy=69.0%, validation loss=0.885
Actual output	RVUMR-14: Training Accuracy: 90.5%, Validation Accuracy: 95.2%, Validation Loss: 0.181
Remarks	This study by validating from a benchmarked dataset we can conclude that our dataset is accurate and valid. Hence, can be used for further research matters.

Table 7.9 System Testing

The third system test (shown in Table 7.9) is for the comparison of both the datasets and hence, validating the same.

7.5 Summary

Software testing is the focus of the whole chapter. It all starts with unit testing all of the software's modules. Then it develops the integration testing phase by evaluating the interface between the modules. Finally, it outlines the system testing phase, which verifies the software's overall functionality.

CHAPTER 8-
EXPERIMENTAL RESULTS AND ANALYSIS

CHAPTER 8

Experimental Results and Analysis

8.1 Evaluation Metrics

The project “Synthetic underwater mine dataset generation” involves a comparative study of RVUMR-14 dataset and a benchmarked dataset CIFAR-10. Both loaded onto the CNN model to determine the accuracy and precision of the model in predicting the same.Hence, validating our custom dataset, if obtained good results.further a prediction model used to classify the test dataset over multiple Evaluation metrics like accuracy, precision, recall ,f1-score etc.,

8.2 Experimental Dataset

Given the nature of the problem statement, there is a scarcity of sea mine image data. Hence acustom dataset is formulated. In order to overcome this obstacle of lack of public availability of data, image augmentation is performed on the initial few images to rapidly multiply the amount of samples available for testing and training. The dataset used in this project consists of underwater mine images. The RVUMR-14 dataset is created by applying augmentation techniques to 14 original images, resulting in a total of 2100 images. These augmentation techniques include rotations, translations, flips, and color transformations.In the fields of computer vision and machine learning, the CIFAR-10 dataset is a frequently used benchmark dataset. It has 60,000 color images that are each 32x32 pixels in size and is divided into 10 classes. 10,000 test photos and 50,000 training images make up the dataset.The CIFAR-10 dataset's 10 classes represent a variety of items and creatures, including trucks, cars, horses, frogs, deer, cars, birds, cats, and cats. A balanced dataset is produced because there are an equal number of photos in each class.

Common uses of the CIFAR-10 dataset include picture classification, object recognition, and feature extraction. Its small image size and diverse object categories make it a suitable dataset for testing and benchmarking different machine learning algorithms and models.The dataset is publicly available and has become a standard benchmark for evaluating the performance of various computer vision algorithms and models. It has played a significant role in advancing the field of deep learning, enabling researchers to develop and compare state-of-the-art methods for image classification and related tasks.

Due to its popularity and accessibility, the CIFAR-10 dataset has served as a foundation for many research papers, competitions, and educational materials in the field of computer vision and machine learning.

8.3 Performance Analysis

Both the datasets are trained on the same CNN model. Where the RVUMR-14 dataset gives an accuracy of 91% and the CIFAR-10 dataset gives accuracy of 76%. Both the models are very good in predicting images and visualize the output.

Table 8.3.1 RVUMR-14 dataset training with CLAHE

SL.NO	FINAL ITER	BATCH SIZE	EPOCH	TRAINING ACCURACY	VALIDATION ACCURACY	VALIDATION LOSS
0	1000	21	13	76.2	81	0.676
1	1500	21	19	90.5	95.2	0.181
2	2000	21	26	100	81	0.323
3	1000	32	20	93.8	93.8	0.284
4	1500	32	29	100	75	0.619
5	2000	32	39	100	71.9	1.047
6	1000	42	26	100	76.2	0.748
7	1500	42	38	95.2	88.1	0.784
8	2000	42	51	97.6	78.6	0.800
9	1000	15	9	86.7	73.6	0.980

Table 8.3.2 RVUMR-14 dataset training without CLAHE

SL.NO	FINAL ITER	BATCH SIZE	EPOCH	TRAINING ACCURACY	VALIDATION ACCURACY	VALIDATION LOSS
0	1000	15	9	86.7	73.3	0.812
1	1000	21	13	100	61.9	1.119
2	1500	21	19	95.2	85.7	0.464
3	2000	21	26	100	85.7	0.722
4	1000	32	20	84.4	75	0.565
5	1500	32	29	100	84.4	0.575
6	2000	32	29	100	87.5	0.322
7	1000	42	26	95.2	73.8	0.774
8	1500	42	38	100	83.1	0.572
9	2000	42	51	100	83.3	0.519

Table 8.3.3 RVUMR-14 dataset prediction

SL.NO	IMAGE	PREDICT	F1_SCORE	MSE	PRECISION /RECALL
0	AR0	AR0	1.0	0.0	[1.0,1.0]
1	HF1	IG3	0.0	0.1428	[0.923,0.0]
2	HG7	HG7	1.0	0.0	[1.0,1.0]
3	IG3	HF1	0.0	0.1428	[0.923,0.0]
4	MF9	MF9	1.0	0.1428	[0.923,0.0]
5	MGB12	MGB12	1.0	0.0714	[0.923,0.0]
6	MU2	MU2	1.0	0.1428	[0.923,0.0]
7	MU5	MU5	1.0	0.1428	[0.923,0.0]
8	MU6	MU6	1.0	0.1428	[0.923,0.0]
9	HG7	HG7	0.0	0.1428	[0.923,0.0]
10	MU10	MU10	1.0	0	[0.923,0.0]
11	MU6	MU6	0.0	0.1428	[0.923,0.0]
12	SGB11	SGB11	1.0	0	[1.0,1.0]
13	TG13	TG13	1.0	0	[1.0,1.0]

Table 8.3.4 CIFAR-10 dataset training with CLAHE

SL.NO	FINAL ITER	BATCH SIZE	EPOCH	TRAINING ACCURACY	VALIDATION ACCURACY	VALIDATION LOSS
0	20000	21	11	57.1	61.9	1.037
1	25000	21	14	57.1	71.4	1.130
2	30000	21	16	76.2	61.9	0.944
3	20000	32	17	53.1	62.5	1.131
4	25000	32	21	50	62.5	1.318
5	30000	32	24	71.9	56.2	1.392
6	20000	42	22	66.7	59.5	1.203
7	25000	42	27	64.3	57.1	1.154
8	30000	42	32	61.9	71.4	0.385
9	35000	42	37	81.0	69.0	0.885
10	40000	42	43	73.8	59.5	1.229

Table 8.3.5 CIFAR-10 dataset training without CLAHE

SL.NO	FINAL ITER	BATCH SIZE	EPOCH	TRAINING ACCURACY	VALIDATION ACCURACY	VALIDATION LOSS
0	20000	21	11	76.2	85.7	0.754
1	25000	21	14	76.2	52.4	1.129
2	30000	21	16	61.9	52.4	1.294
3	20000	32	17	56.2	65.6	0.860
4	25000	32	21	65.6	68.8	0.816
5	30000	32	24	71.9	62.5	1.258
6	20000	42	22	66.7	52.4	1.567
7	25000	42	27	73.8	64.3	1.084
8	30000	42	32	78.6	59.5	1.128
9	15000	21	8	66.7	61.9	1.030

Table 8.3.6 CIFAR-10 dataset prediction model

SL.NO	IMAGE	PREDICTED	F1_SCORE	MSE	PRECISION /RECALL
0	AIRPLANE	AIRPLANE	1.0	0.0	[1.0,1.0]
1	AUTOMOBILE	AUTOMOBILE	1.0	0.0	[1.0,1.0]
2	BIRD	BIRD	1.0	0.0	[1.0,1.0]
3	CAT	CAT	1.0	0.0	[1.0,1.0]
4	DEER	CAT	0.00	0.2	[0.888,0.0]
5	DOG	DOG	1.0	0.0	[1.0,1.0]
6	FROG	BIRD	0.0	0.2	[0.888,0.0]
7	HORSE	CAT	0.0	0.2	[0.888,0.0]
8	SHIP	SHIP	1.0	0.0	[1.0,1.0]
9	TRUCK	TRUCK	1.0	0.0	[1.0,1.0]

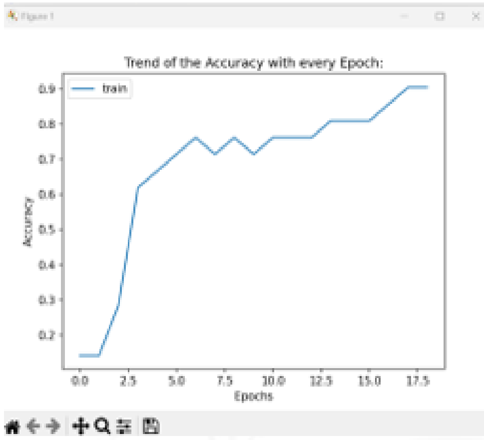


Fig 8.3.7 Trend of accuracy for training with CLAHE(RVUMR-14)

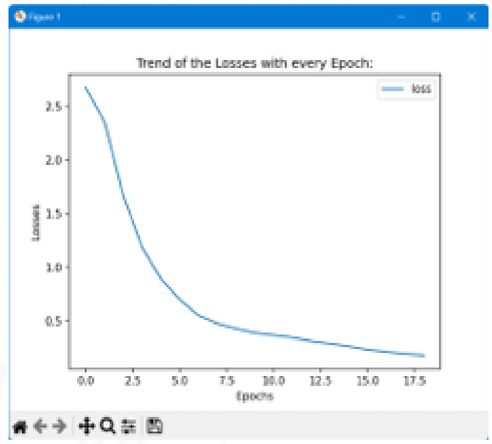


Fig 8.3.8 Trend of loss for training with CLAHE(RVUMR-14)

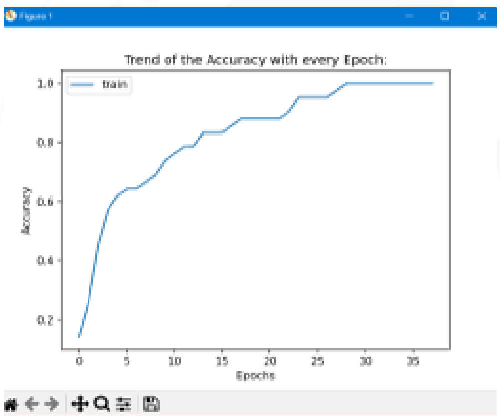


Fig 8.3.9 Trend of accuracy for training without CLAHE(RVUMR-14)

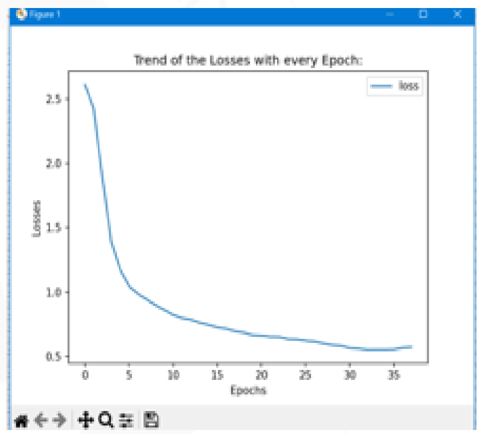


Fig 8.3.10 Trend of loss for training without CLAHE(RVUMR-14)

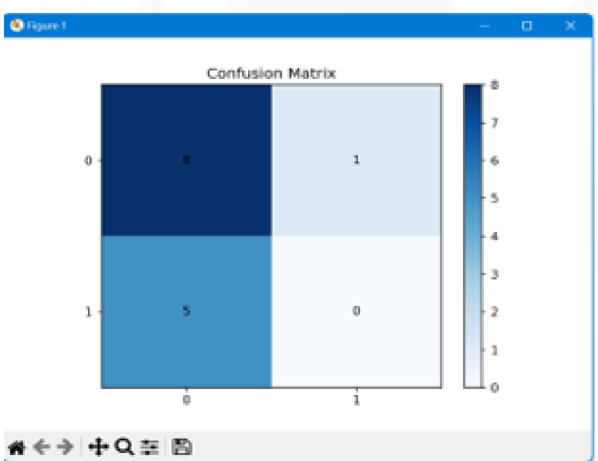


Fig 8.3.11 confusion matrix of input images for prediction

Fig 8.3.12 Trend of accuracy for training with CLAHE(CIFAR- 10)

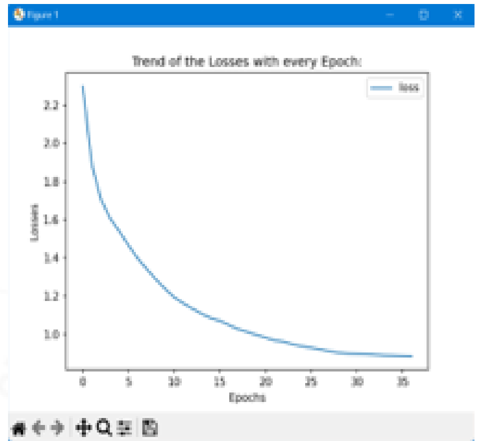


Fig 8.3.13 Trend of loss for training with CLAHE(CIFAR-10)

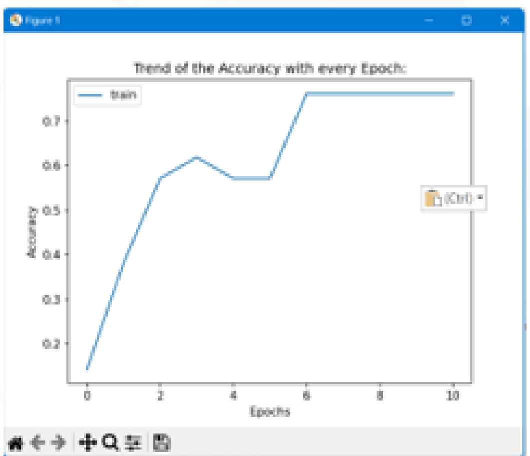


Fig 8.3.14 Trend of accuracy for training without CLAHE(CIFAR-10)

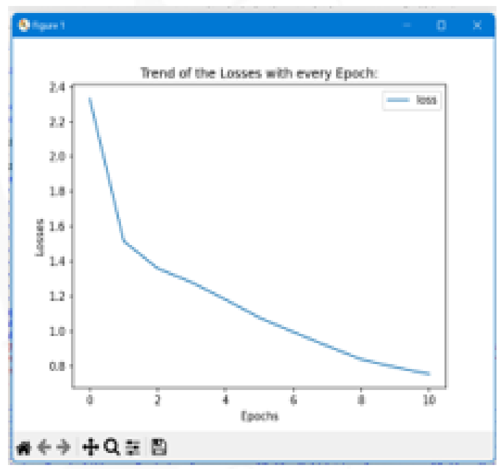


Fig 8.3.15 Trend of loss for training without CLAHE(CIFAR-10)

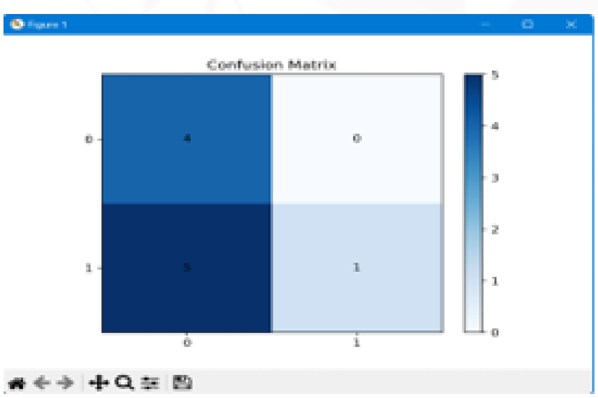


Fig 8.3.16 confusion matrix of input images for prediction

8.4 Summary

The results were useful to establish which machine learning technique is most suitable for the identification of underwater marine mines. The best-suited classification machine-learning technique focuses mostly on three measures – precision, retrieval, and f1-score. There were no flaws or mistakes in the system. At scaling, the system is stable. The entire system performance is hence efficient.

CHAPTER 9 – CONCLUSION AND FUTURE ENHANCEMENTS

CHAPTER 9

Conclusion and Future Enhancements

9.1 Limitations of the Project

This project has several limitations to consider. Firstly, the dataset size, consisting of 2100 images generated from 14 originals, may not fully capture the diversity and complexity of real-world underwater mine scenarios. Additionally, the dataset's labeling accuracy can impact the model's performance and the correctness of the dataset evaluation. The choice of a specific CNN model for underwater mine detection may limit the exploration of alternative architectures. Furthermore, while the project focuses on dataset creation and model evaluation, challenges related to real-world deployment and integration into operational systems are not addressed. Finally, the effectiveness of the augmented dataset is evaluated based on comparison with a benchmarked dataset, which may not fully encompass the range of real-world variations and challenges.

9.2 Future Enhancements

In future enhancements, the project can focus on increasing the dataset size to capture more underwater mine variations. Fine-grained annotations, such as segmenting individual mines, can be added for advanced analysis. Integration of multi-modal data like acoustic or sonar information can enhance detection accuracy. Real-time deployment considerations can be addressed, optimizing the models for operational systems. Transfer learning and model fine-tuning using pre-trained models can improve generalization. Collaboration with other researchers and institutions for dataset exchange and benchmarking efforts can lead to diverse perspectives and advancements. These enhancements will contribute to more accurate and reliable underwater mine detection systems.

9.3 Summary

In summary, this project aims to address the limitations of underwater mine detection by creating an augmented dataset using augmentation techniques and evaluating its effectiveness compared to a benchmarked dataset. The dataset consists of 2100 images generated from 14 original images, with preprocessing using CLAHE to enhance image quality. A CNN model is employed for underwater mine detection. The project's future enhancements include increasing the dataset size, incorporating fine-grained annotations and multi-modal data, addressing real-time deployment challenges, exploring transfer learning, and fostering collaboration with other researchers. These enhancements aim to improve the accuracy and reliability of underwater mine detection systems and contribute to advancements in the field

9. REFERENCES

[1] Abhishek, N., et al. "Underwater detection of mines using image processing." International Research Journal of Engineering and Technology, vol. 7, no. 5, 2020. IRJET, https://www.irjet.net/.

[2] Agarwal, Aditya, et al. "Underwater Fish Detection." International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 4, 2020, p. 4. IJERT,https://www.ijert.org/underwater-fish-detection.

[3] DENOS, Killian, et al. "Deep Learning applied to Underwater Mine Warfare."
Institute of Electrical and Electronics Engineers, 2017, pp. 1-7.

[4] Galceran, Enric, et al. A Real-time Underwater Object Detection Algorithm forMulti-beam Forward Looking Sonar. 2012. Sciencedirect, https:// www.sciencedirect.com/science/article/pii/S1474667016306206.

[5] Geethalakshmi, S.N., et al. "A Study on Detection and Classification of UnderwaterMines using Neural Networks." International Journal of Soft Computing and Engineering (IJSCE), vol. Volume - 1, no. Issue - 5, 2011, p. 8. IJSCE, https:// zwww.ijsce.org/wp- content/uploads/papers/v1i5/E0173091511.pdf.

[6] K, Muthukaran, and Merlin Moses M. "Color Image Segmentation using K-means Clustering and Optimal Fuzzy C-Means Clustering." International Conference onCommunication and Computational Intelligence (INCOCCI), 2010. IEEE Xplore,https://ieeexplore.ieee.org/abstract/document/5738735.

[7] Mishra, Ishan, and Laurens Maaten. Self-Supervised Learning of Pretext-Invariant Representations. 2019, https://www.researchgate.net/publication/337756774_Self- Supervised_Learning_of_Pretext-Invariant_Representations.

[8] Neupane, Dhiraj, and Jongwon Seok. "A Review on Deep Learning-Based Approaches for Automatic Sonar Target Recognition." MDPI Electronics, vol. 9, no. 11, 2020, p. 1.

[9] Nguyen, Huu-Thu, et al. "Study on the Classification Performance of Underwater Sonar Image Classification Based on Convolutional Neural Networks for Detecting aSubmerged Human Body." MDPI Sensors, vol. 20, no. 1, 2019, p. 1. MDPI Sensors,https://www.mdpi.com/2079-9292/9/11/1972.

[10] Premana, Agyztia, et al. "Image Segmentation Using Gabor Filter and K-Means Clustering Method." International Seminar on Application for Technology of Information and

Communication (iSemantic), 2017, pp. 95-99. IEEE Xplore, https://ieeexplore.ieee.org/document/8251850.

[11] S, Manonmani, et al. “Survey On Image Denoising Techniques.” International Journal of Science, Engineering and Technology Research, vol. 5, no. 9, 2016. ijestr, www.ijestr.com.

[12] Shrivastava, Ashish, et al. “Learning from Simulated and Unsupervised Images through Adversarial Training”. 2016. https://arxiv.org/abs/1612.07828, https://arxiv.org/abs/1612.07828.

[13] Thakre, Shiwam S., and Amir Sahu. “Comparative Analysis of Various Underwater Image Enhancement Techniques.” International Journal of Computer Science and Mobile Computing, vol. 3, no. 4, 2014, pp. 33-88. IJCSMC, www.ijcsmc.com.

[14] Zheltonozhskii, Evgenii, et al. “Self-Supervised Learning for Large-Scale Unsupervised Image Clustering.” arXiv.org, 2020. arXIV.org, https://arxiv.org/abs/2008.10312

[15] Rios, John. (2005). Naval Mines in the 21st Century: Can NATO Navies Meet the Challenge?. 79.

[16] Tellez, O.. “Human-in-the-loop for autonomous underwater threat recognition.” OCEANS 2018 MTS/IEEE Charleston (2018): 1-5.

[17] Tellez, O.. “Underwater threat recognition: Are automatic target classification algorithms going to replace expert human operators in the near future?” OCEANS 2019 - Marseille (2019): 1-4.

[18] Li, Sichun, Xin Jin, S. Yao and Shuyu Yang. “Underwater Small Target Recognition Based on Convolutional Neural Network.” Global Oceans 2020: Singapore – U.S. Gulf Coast (2020): 1-7.

[19] Cho, Hyeonwoo, Juhyun Pyo, J. Gu, Hangil Jeo and Son-cheol Yu. “Sonar simulator-based underwater object recognition and its tolerance evaluation.” OCEANS 2016 - Shanghai (2016): 1-7.

[20] Teng, B. and Hongjian Zhao. “Underwater target recognition methods based on the framework of deep learning: A survey.” International Journal of Advanced Robotic Systems 17 (2020): n. pag.

[21] Cao, X., X. Zhang, Yang Yu and Letian Niu. “Deep learning-based recognition of underwater target.” 2016 IEEE International Conference on Digital Signal Processing (DSP) (2016): 89-93.

[22] Yu, Xian, Xiangrui Xing, Han Zheng, Xueyang Fu, Yue Huang and Xinghao Ding. “Man-Made Object Recognition from Underwater Optical Images Using Deep Learning and Transfer Learning.” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2018): 1852-1856.

[23] Hu, Gang, Kejun Wang and L. Liu. “Underwater Acoustic Target Recognition Based on Depthwise Separable Convolution Neural Networks.” Sensors (Basel, Switzerland) 21 (2021): n. pag.

[24] Cao, X., R. Togneri, Xiaomin Zhang and Yang Yu. "Convolutional Neural Network With Second-Order Pooling for Underwater Target Classification." IEEE Sensors Journal 19 (2019): 3058-3066.

[25] Kamal, S., S. K. Mohammed, P. Pillai and M. Supriya. "Deep learning architectures for underwater target recognition." 2013 Ocean Electronics (SYMPOL) (2013): 48- 54.

[26] Hu, Gang, K. Wang, Yuan Peng, Mengran Qiu, Jianfei Shi and L. Liu. "Deep Learning Methods for Underwater Target Feature Extraction and Recognition.

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

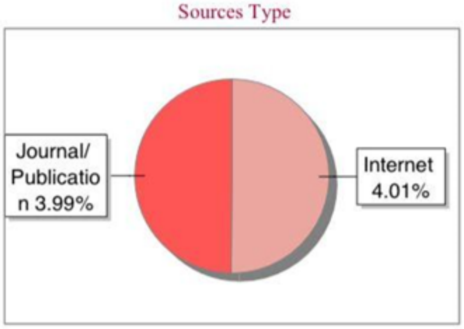
Submission Information

Author Name	moin
Title	report3
Paper/Submission ID	756519
Submission Date	2023-05-30 10:45:42
Total Pages	101
Document type	Project Work

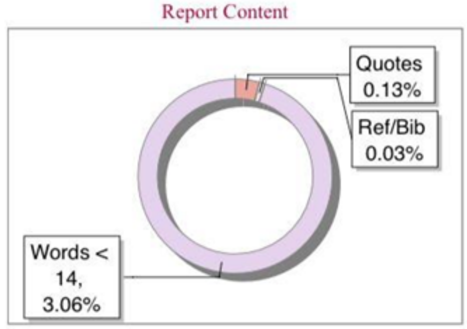
Result Information

Similarity **8 %**

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

8	98	A	A-Satisfactory (0-10%)
SIMILARITY %	MATCHED SOURCES	GRADE	B-Upgrade (11-40%)
			C-Poor (41-60%)
			D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	dspace.daffodilvarsity.edu.bd 8080	<1	Publication
2	ijeat.org	<1	Publication
3	journals.plos.org	<1	Internet Data
4	plosjournal.deepdyve.com	<1	Internet Data
5	Article Published By Asia-Pacific Journal Japan Focus Journal - www.apjjf.org	<1	Publication
6	moam.info	<1	Internet Data
7	qdoc.tips	<1	Internet Data
8	Deep learning of brain magnetic resonance images A brief review by Zhao-2020	<1	Publication
9	Thesis Submitted to Shodhganga Repository	<1	Publication
10	www.science.gov	<1	Internet Data
11	An automatic approach to the detection and extraction of mine features in sidesc by Reed-2003	<1	Publication
12	mdpi.com	<1	Internet Data
13	mdpi.com	<1	Internet Data

14	artsdocbox.com	<1	Internet Data
15	Detecting cerebral microbleeds with transfer learning by Hong-2019	<1	Publication
16	www.researchgate.net	<1	Internet Data
17	www.iraj.in	<1	Publication
18	moam.info	<1	Internet Data
19	IEEE 2014 IEEE 10th International Colloquium on Signal Processing by	<1	Publication
20	en.unionpedia.org	<1	Internet Data
21	moam.info	<1	Internet Data
22	discovher.life	<1	Internet Data
23	Scaling up the Synthesis of a Hydroxyquinoline-Functionalized p-tert-B by Roode-Gutzmer-2019	<1	Publication
24	SOLOMO Are hospitality SMEs able to move beyond traditional websites in their by Kriechbaumer-2019	<1	Publication
25	www.researchgate.net	<1	Internet Data
26	qdoc.tips	<1	Internet Data
27	www.ijert.org	<1	Internet Data
28	www.atlantis-press.com	<1	Publication
29	its.papercept.net	<1	Internet Data
30	www.readbag.com	<1	Internet Data
31	abnews-wire.blogspot.com	<1	Internet Data

32	A study on different closed domain question answering approaches by Badugu-2020	<1	Publication
33	moam.info	<1	Internet Data
34	s-space.snu.ac.kr	<1	Publication
35	IEEE 2014 IEEE 10th International Colloquium on Signal Processing by	<1	Publication
36	moam.info	<1	Internet Data
37	www.dx.doi.org	<1	Publication
38	aclweb.org	<1	Internet Data
39	Fast training of multilayer perceptrons by Verma-1997	<1	Publication
40	Fish metacommunity responses to experimental drought are determined by habitat h by Driver-2016	<1	Publication
41	index-of.es	<1	Publication
42	mdpi.com	<1	Internet Data
43	uir.unisa.ac.za	<1	Publication
44	www.frontiersin.org	<1	Publication
45	adoc.pub	<1	Internet Data
46	Thesis Submitted to Shodhganga Repository	<1	Publication
47	Validation of an E-cigarette Purchase Task in Advanced Generation Device Users by Cassidy-2020	<1	Publication
48	www.freepatentsonline.com	<1	Internet Data
49	www.wimsoft.com	<1	Internet Data

50	arxiv.org	<1	Publication
51	erj.ersjournals.com	<1	Publication
52	IEEE 2018 20th International Symposium on Electrical Apparatus and , by Hinov, Nikolay Hra- 2018	<1	Publication
53	www.dx.doi.org	<1	Publication
54	www.readkong.com	<1	Internet Data
55	ijrte.org	<1	Publication
56	mdpi.com	<1	Internet Data
57	IEEE 2014 IEEE Workshop on Biometric Measurements and Systems for Se by	<1	Publication
58	academiccommons.columbia.edu	<1	Publication
59	Estimation of potential zones for offshore mariculture in the Indian Sea using g by Jha-2017	<1	Publication
60	Market power and firm risk A test of the quiet life hypothesis by Stephe-1982	<1	Publication
61	mdpi.com	<1	Internet Data
62	moam.info	<1	Internet Data
63	www.wtec.org	<1	Internet Data
64	coek.info	<1	Internet Data
65	moam.info	<1	Internet Data
66	North Bengal University Thesis Published in - www.inflibnet.ac.in	<1	Publication
67	philpapers.org	<1	Internet Data

68	Predicting novel substrates for enzymes with minimal experimental effort with ac by Pertusi-2017	<1	Publication
69	ro.uow.edu.au	<1	Publication
70	www.intechopen.com	<1	Publication
71	www.mdpi.com	<1	Internet Data
72	www.sciencegate.app	<1	Internet Data
73	mdpi.com	<1	Internet Data
74	A Structured Linear Programming Model in the Food Industry by Williams-1974	<1	Publication
75	drishtiias.com	<1	Internet Data
76	fdocuments.in	<1	Internet Data
77	IEEE 2014 IEEE 10th International Colloquium on Signal Processing by	<1	Publication
78	jurnal.unej.ac.id	<1	Publication
79	Precuneus stimulation alters the neural dynamics of autobiographical memory retr by Hebscher-2020	<1	Publication
80	Thesis Submitted to Shodhganga, shodhganga.inflibnet.ac.in	<1	Publication
81	The Excimer Laser Current Knowledge and Future Prospects by JAME-1988	<1	Publication
82	www.intechopen.com	<1	Publication
83	APPROVED EXAMINING BOARDS IN MEDICAL SPECIALTIES by Kirklin-1952	<1	Publication

84	A study of the stability of the rolling motion of a ship in an irregular seaway by Muhuri-1980	<1	Publication
85	Deep learning in bioinformatics Introduction, application, and persp, by Li, Yu Huang, Chao- 2019	<1	Publication
86	docplayer.net	<1	Internet Data
87	documents.mx	<1	Internet Data
88	dokumen.pub	<1	Internet Data
89	ijitee.org	<1	Internet Data
90	index-of.es	<1	Publication
91	journal.gnest.org	<1	Publication
92	moam.info	<1	Internet Data
93	moam.info	<1	Internet Data
94	moam.info	<1	Internet Data
95	sti.bmj.com	<1	Publication
96	Thesis Submitted to Shodhganga Repository	<1	Publication
97	www.amsa.gov.au	<1	Internet Data
98	www.mecs-press.org	<1	Publication

Synthetic underwater mine dataset generation using deep neural networks

Mohamed Moin Irfan, *Dept of CSE*
RV College of Engineering, Bangalore, India
Mohamedmoini.cs19@rvce.edu.in

Prerana Shekar M S *Dept of CSE*
RV College of Engineering, Bangalore, India
Preranashekarms.cs18@rvce.edu.in

Purnodeep Ranjankar *Dept of CSE*
RV College of Engineering, Bangalore, India
purnodeepr.cs19@rvce.edu.in

Sathvik Gowda M, *Dept of CSE*
RV College of Engineering, Bangalore, India
sathvikgowdam.cs18@rvce.edu.in

Prof Manonmani S
Assistant Professor
Dept of CSE
RV College of Engineering,India
manonmanis@rvce.edu.in

***Abstract*-This paper presents the implementation details and performance evaluation of RVUMR-14, a custom dataset specifically designed for underwater mine classification. Due to limited availability of images, various augmentation techniques were employed to increase the dataset size. The dataset consists of fourteen different types of underwater mines, with 150 images in each class. Further, annotated to get masks with labels for any future semantic segemtation applications. The The Adam optimizer was employed during training, while the Contrast Limited Adaptive Histogram Equalisation (CLAHE) technique was used for preprocessing. The dataset was used to train a convolutional neural network (CNN) model, with training and validation split 80:20. The model had a 91% accuracy rate. RVUMR-14's efficacy is verified by a comparison that was made with the benchmarked CIFAR-10 dataset, where RVUMR-14 outperformed CIFAR-10, obtaining an accuracy of 91% compared to 76%.**

1.Introduction

Underwater mine detection play a crucial role in maritime security, underwater exploration, and defense operations. Accurate identification and classifying of underwater mines are essential for ensuring the safety of naval vessels, marine life, and human divers. However, this task is challenging because of the complex nature of underwater environments, limited visibility, and the wide variety of mine types. Creating custom datasets for underwater mine classification is vital as publicly available datasets specific to this domain are limited. The availability of diverse and annotated underwater mine images is crucial to train robust machine learning models that can effectively classify different types of mines.

In this context, the present study introduces RVUMR-14, a custom dataset developed specifically for underwater mine classification. The dataset aims to address the scarcity of publicly available underwater mine images and provide a comprehensive and diverse collection of annotated images for training and evaluation purposes. The RVUMR-14 dataset comprises images of 14 distinct types of underwater mines commonly encountered in maritime environments. Different augmentation strategies were used to enhance the number of photos per class in order to get over the drawback of a tiny dataset. These techniques included random rotations, flips, and translations, which simulate variations in mine appearance and positioning. To enhance the quality of the underwater mine images and improve the discriminative features, the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm was applied as a preprocessing step. CLAHE enhances the contrast and details in the images, making them more suitable for subsequent feature extraction and classification.

The CNN model was chosen as the underlying architecture for underwater mine classification due to its ability to effectively capture spatial dependencies and extract discriminative features from images. The model architecture was carefully designed, consisting of multiple convolutional and pooling layers, followed by fully connected layers for classification. The RVUMR-14 dataset was divided into an 80% training set and a 20% validation set. During the training process, the Adam optimizer, known for its efficiency in training deep learning models, was utilized to optimize the parameters in the CNN model. The model was trained until convergence or until a predefined stopping criterion was met. The performance of the trained CNN model was evaluated on a separate test set, which included previously unseen underwater mine images. Various evaluation metrics, including F1 score, Mean Squared Error (MSE), confusion matrix, precision, and recall, were employed to assess the model's classification accuracy and overall performance. To validate the effectiveness of the RVUMR-14 dataset, a comparison was made with the benchmarked CIFAR-10 dataset, which represents a general image classification task. The comparison revealed that the RVUMR-14 dataset outperformed CIFAR-10, demonstrating its effectiveness in accurately classifying underwater mines.

In summary, this paper introduces the RVUMR-14 dataset and presents the detailed implementation of a CNN

Keywords: convolutional neural network, mine countermeasure, Adam optimizer.

2. Dataset Creation

2.1 Selection of Underwater Mine Types:
The 1st step in creating the RVUMR-14 dataset was to identify and select the various types of underwater mines to be included. This process involved consulting domain experts, studying existing literature on underwater mine classification, and considering the diversity of mines encountered in real-world scenarios. A comprehensive list of 14 distinct mine types was compiled, representing a range of shapes, sizes, and materials commonly found in underwater minefields.

2.2 Data Acquisition:
Acquiring a sufficient number of underwater mine images for each mine type is challenging due to the limited availability of publicly accessible datasets. To overcome this limitation, we collected 14 different types of mines and gave them a code name. The code name consist of prt of its name, the country of origin and the grayscale value corresponding to its labelled colour index. Further, these images were augmented.

2.3 Augmentation Techniques:
Due to the limited number of available images for each mine type, augmentation techniques were employed to increase the dataset size and introduce variations in the images. Augmentation helps enhance the model's ability to generalize and perform well on unseen data. Various augmentation techniques were used to the collected images. Random rotations within a specified range were applied to simulate different orientations of the mines. Flips (horizontal and vertical) were performed to introduce mirror images of the mines. Translations were applied to simulate variations in mine position within the image frame. These augmentation techniques significantly expanded the dataset, resulting in a bigger and more diverse collection of underwater mine images.

2.4 Annotation and Labeling:
Once the dataset was assembled, the next step was to annotate and label the images. Each image was carefully examined, and pixel-level segmentation masks were created to indicate the location of the mine within the image. These annotations provided ground truth info for training and evaluating for future research in the semantic segmentation.

2.5 Dataset Composition:
The final RVUMR-14 dataset consisted of 14 various types of underwater mines, with 150 images per mine type. This contrinuted in a total of 2,100 annotated images. The dataset encompassed a diverse range of mine shapes, sizes, materials, and orientations, reflecting the challenges encountered in real-world underwater mine classification scenarios. To ensure unbiased model evaluation and prevent data leakage, the dataset was stratified and randomly split into an 80% training set and a 20% validation set. The stratification ensured that each mine type was represented proportionally in both training and validation sets, preserving the dataset's class distribution.

The RVUMR-14 dataset, with its annotated underwater mine images and corresponding labels, provides a valuable resource for training and evaluating CNN models specifically designed for underwater mine classification. The dataset's composition, diversity, and annotations enable researchers and practitioners to develop and validate accurate and robust models for underwater mine detection and classification tasks.

3. Preprocessing

Preprocessing plays a crucial role in preparing the RVUMR-14 dataset for effective feature extraction and classification. In this implementation, the CLAHE algorithm was employed as a preprocessing technique to enhance the image quality and improve the visibility of underwater mines.

3.1 Image Enhancement with CLAHE:
Underwater imagery often suffers from low visibility, low contrast, and uneven illumination due to factors such as water turbidity, light attenuation, and scattering. These challenges can hinder the accurate detection and identification of underwater mines. To mitigate these issues, the CLAHE

algorithm was used to enhance the images before loading them into the CNN model.

CLAHE is an adaptive contrast enhancement algorithm that improves image quality by redistributing the pixel intensities to achieve a more balanced and enhanced contrast. The algorithm operates by dividing the image into small, overlapping subregions called tiles. Within each tile, a histogram equalization process is applied to stretch the intensity values to a wider range while preserving local contrast. This adaptive approach prevents over-enhancement and maintains the natural appearance of the images. By applying CLAHE, the contrast and visibility of underwater mine images are significantly improved. The algorithm enhances the details and textures, making the mines more distinguishable and facilitating better feature extraction during the subsequent stages of the classification process.

Fig 3.1: Image without pre-processing

Fig 3.2: Image with pre-processing(CLAHE)

3.2 Additional Preprocessing Steps:
In addition to CLAHE, other preprocessing steps may be applied depending on the specific characteristics of the RVUMR-14 dataset and the requirements of the CNN model. Some common preprocessing techniques used in underwater image analysis include:

3.2.1 Color Space Conversion:
Underwater images captured in RGB color space may suffer from color distortions due to water absorption and scattering. Converting the images to alternative color spaces such as Lab or HSV can help mitigate these distortions and provide more reliable color information for classification.

3.2.2 Image Resizing and Cropping:
Resizing the images to a consistent resolution can facilitate efficient model training and reduce computational complexity. Additionally, cropping the images to focus on the region of interest (i.e., the mine) can further enhance classification performance by reducing irrelevant background information.

3.2.3 Noise Reduction:
Underwater images are often affected by various types of noise, including salt-and-pepper noise and Gaussian noise. Applying denoising techniques such as median filtering or Gaussian smoothing can help reduce the noise levels and improve the clarity of the images. It is to note that the selection and application of preprocessing techniques may vary depending on the specific characteristics of the RVUMR-14 dataset, the nature of the underwater mine images, and the requirements of the CNN model. Experimentation and fine-tuning of the preprocessing steps are necessary to achieve optimal results.

By applying CLAHE and potentially other preprocessing techniques, the RVUMR-14 dataset is preprocessed to enhance image quality, improve visibility, and provide a more suitable input for the CNN model. These preprocessing steps enable the model to effectively extract relevant features and accurately classify underwater mines during the subsequent stages of training and evaluation.

4. CNN Model Architecture

The Convolutional Neural Network architecture plays a useful role in extracting meaningful features from the preprocessed underwater mine images and performing accurate classification. In this implementation, a carefully designed CNN model was utilized to obtain high classification accuracy for the RVUMR-14 dataset.

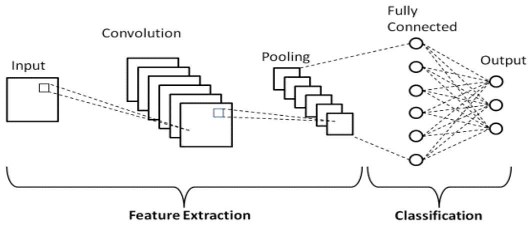


Fig 4.1: CNN architecture.

4.1 Convolutional Layers:

The CNN model begins with a number of convolutional layers that specialize in extracting local features and spatial information from the input images. Each convolutional layer consists of multiple filters or kernels that convolve with the input to produce feature maps. The filters learn to detect specific patterns, such as edges, textures, and shapes, at different scales and orientations.

The quantity of convolutional layers used affects how intricate and complex the CNN architecture is. Although they can need more computational resources, deeper architectures with more convolutional layers frequently capture more abstract and high-level characteristics. To enable the model to learn more complicated representations, the number of filters in each layer can be incrementally raised.

4.2 Pooling Layers:

In order to decrease the spatial dimensions of the feature maps and abstract away the precise spatial information, pooling layers are often included between subsequent convolutional layers. The most typical pooling process is known as max pooling, which chooses the highest value possible within a predetermined pool size. Max pooling helps in reducing the computational burden, providing a form of spatial invariance, and promoting translation invariance.

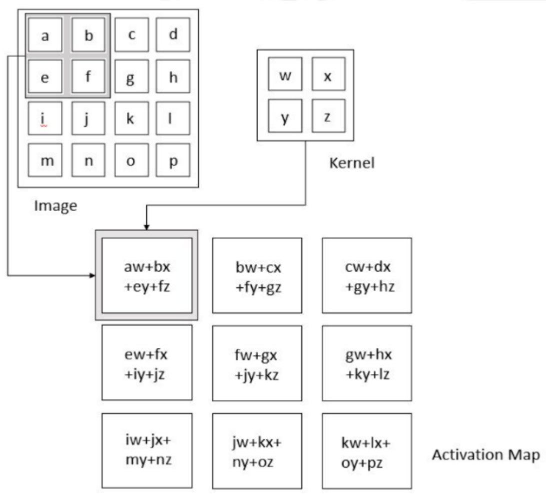


Fig 4.2: Activation Map

4.3 Activation Functions:

Activation functions introduce non-linearities into the CNN model, enabling it to learn complex decision boundaries and capture non-linear relationships between the features and class labels. Common activation functions used in CNN models include Rectified Linear Units (ReLU), sigmoid, and hyperbolic tangent (tanh). ReLU is widely preferred due to its simplicity, computational efficiency, and avoidance of the vanishing gradient problem.

4.4 Fully Connected Layers:

The collected features are run through numerous convolutional and pooling layers before being fed via fully connected layers. These layers are in charge of discovering the overarching trends and connections between the extracted features. Each neuron in the layer with complete connectivity is linked to every neuron in the layer below. The last completely connected layer normally has the same number of neurons as the total number of classes in the classification job, though the number of neurons in the fully connected layers can vary.

4.5 Dropout and Regularization:

Regularisation strategies are used to avoid overfitting, which happens when the model performs well on the training data but is unable to generalise to new data. Dropout is a regularisation method that is frequently used. During training, a random subset of neurons in the fully connected layers are "dropped out" for a brief period of time. This stops neurons from co-adapting and helps the network to learn stronger features.

4.6 Output Layer and Activation:

The output layer, which is the last layer of the CNN model, is made up of neurons that correspond to the number of classes in the classification task. Class probabilities, which reflect the model's level of confidence in each class prediction, are generated for multi-class classification using a softmax activation function. For a particular input image, the class with the highest probability is regarded as the predicted class.

Convolutional layers, pooling layers, fully connected layers, activation functions, and output layers make up the majority of the CNN model architecture. Through experimentation and architectural design considerations, the precise arrangement of these layers—including the number of layers, filter sizes, pooling sizes, and number of neurons—is established. The objective is to create a model that can accurately identify underwater mines in the RVUMR-14 dataset and efficiently extract pertinent features from the input photos.

5. Training and Validation

Once the CNN model architecture is defined and the RVUMR-14 dataset is preprocessed, the next step is to train and validate the model. This involves splitting the dataset into training and validation sets, defining training parameters, and monitoring the model's performance during training.

5.1 Dataset Split:
The RVUMR-14 dataset is divided into an 80% training set and a 20% validation set. The stratified sampling technique is commonly employed to ensure that each mine type is represented proportionally in both sets. This helps prevent bias in model training and evaluation, ensuring that the model learns to generalize across all mine types.

5.2.1 Loss Function:
A suitable loss function is chosen to quantify the discrepancy between the predicted class probabilities and the true labels. For multi-class classification tasks, categorical cross-entropy loss is commonly used. It measures the dissimilarity between the predicted class probabilities and the one-hot encoded ground truth labels.

5.2.2 Optimizer:
An optimizer is selected to update the model's weights based on the computed loss during training. The Adam optimizer, a popular choice for deep learning tasks, is often used due to its efficiency in adapting the learning rate and momentum for each weight in the model.

5.2.3 Learning Rate:
The learning rate determines the step size at which the optimizer updates the model's weights. It is a hyperparameter that needs to be carefully tuned. Too high of a learning rate may result in unstable training, while too low of a learning rate may cause slow convergence. Learning rate scheduling techniques, such as reducing the learning rate over time, can be employed to further enhance training stability and convergence.

5.2.4 Batch Size:
The batch size refers to the number of samples processed in each iteration of the training process. Larger batch sizes can speed up training but require more memory. Smaller batch sizes may introduce more stochasticity in the weight updates but can require more iterations to converge. A suitable batch size is chosen based on the available computational resources and the dataset size.

5.3 Model Training:
The CNN model is trained by iteratively optimizing the weights based on the defined loss function and the training dataset. The following steps are performed during each training iteration:

5.3.1 Forward Pass:
An input batch of images is fed into the model, and the model performs a forward pass. The input images propagate through the layers, and activations are computed at each layer until the final output layer.

5.3.2 Loss Computation:
The predicted class probabilities from the output layer are compared to the true labels using the chosen loss function. The loss value is computed as a measure of the model's performance in predicting the correct classes.

5.3.3 Backward Pass and Weight Update:
Backpropagation is used to calculate the gradients of the loss with respect to the weights of the model. The optimizer then uses these gradients to adjust the weights with the goal of reducing the loss function and enhancing the performance of the model.

5.3.4 Iteration and Epochs:
For a certain number of iterations, sometimes referred to as mini-batches, the aforementioned stages are repeated. One epoch is finished once the entire training dataset has been processed in mini-batches. The model must be trained over a number of epochs in order for it to learn from the data and modify the weights to reduce the loss.

5.4 Validation:
After each epoch or a predefined number of iterations, the trained model is evaluated on the validation set to assess its performance on unseen

data. The following steps are done during the validation process:

5.4.1 Forward Pass on Validation Set:
The validation set images are passed through the trained model, and the predicted class probabilities are obtained.

5.4.2 Evaluation Metrics:
To evaluate the performance of the model, a number of evaluation metrics are calculated, including accuracy, precision, recall, and loss. These measures shed light on the model's overall effectiveness as well as its capacity to categorise underwater mines accurately.
5.4.3 Monitoring and Early Stopping:
The validation metrics are monitored throughout the training process, and the model's performance is assessed. Early stopping techniques can be applied to prevent overfitting. If the validation metrics stop improving or start deteriorating, training can be stopped to prevent the model from becoming overly specialized to the training set.

The CNN model's weights can be repeatedly adjusted in order to maximise the model's precision in classifying underwater mines. This can be done by training the CNN model on the RVUMR-14 dataset and evaluating its performance on the validation set. The model is trained and validated to make sure it can generalise well and function well with new data.

6. Performance Evaluation

After training the CNN model on the RVUMR-14 dataset and validating its performance, a comprehensive performance evaluation is conducted to assess the model's effectiveness in classifying underwater mines. Several evaluation metrics and techniques are utilized to measure the model's performance and give insights into its strengths and weaknesses.

6.1 Test Set:
To evaluate the model's performance on unseen data, a separate test set is prepared. This test set consists of underwater mine images that can be used to assess the model's generalization capabilities and real-world performance.

6.2 Prediction on Test Set:
Predictions are made for each image using the test set and the trained model. Based on its learnt features and classification decision boundaries, the model predicts the class label for each image.

6.3 Evaluation Metrics:
A number of evaluation metrics are employed to gauge how well the model performed on the test set. The following measurements are frequently used:

6.3.1 Accuracy:
Accuracy is a fundamental metric that measures the percentage of correctly classified instances in the test set. It provides an overall assessment of the model's correctness in predicting the mine types.

6.3.2 Precision:
Precision measures the ability of the model to correctly identify positive instances (i.e., correctly classify a mine) out of all instances predicted as positive. It indicates the model's reliability in classifying mines without misclassifying other objects as mines.
6.3.3 Recall (Sensitivity):
Recall, also called as sensitivity or true positive rate, measures the ability of the model to correctly identify positive instances out of all actual positive instances in the test set. It reflects the model's ability to capture all instances of the target class.

6.3.4 F1-Score:
The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance by considering both precision and recall. It is useful when dealing with imbalanced datasets, where the number of instances in different classes varies significantly.

6.3.5 Mean Squared Error (MSE):
MSE is a metric commonly used in regression tasks to measure the average squared difference between the predicted values and the actual values. In the context of classification, MSE can be applied by considering the predicted probabilities as continuous values and the one-hot encoded ground truth labels as target values.

6.3.6 Confusion Matrix:
A confusion matrix is a table that displays the number of accurate and unreliable guesses for each class. It offers thorough explanations of the model's

performance for each class, including the number of true positives, true negatives, and false positives. Additional metrics like specificity and false positive rate can be deduced from the confusion matrix.

6.4 Performance Analysis:

The evaluation metrics obtained from the test set provide valuable information about the model's performance. They help identify strengths, weaknesses, and areas for improvement. By analyzing the confusion matrix and examining specific instances, patterns of misclassifications or challenging cases can be identified, guiding further enhancements to the model or dataset.

Comparing the performance of the CNN model on the RVUMR-14 dataset with the performance on a benchmark dataset like CIFAR-10 allows for a validation of the custom dataset's effectiveness. By observing the differences in accuracy, it can be determined whether the RVUMR-14 dataset exhibits better performance in classifying underwater mines compared to a more generic dataset like CIFAR-10.

Through a thorough performance evaluation, the effectiveness and limitations of the CNN model in classifying underwater mines can be assessed. This analysis provides insights into the model's overall performance and guides future improvements in model architecture, dataset collection, and preprocessing techniques for underwater mine classification

Table 6.1 RVUMR-14 dataset training with CLAHE

Sl No	Final Iter	Batch size	Epoch	Training Accuracy	Validation Accuracy	Validation Loss
0	1000	21	13	76.2	81	0.676
1*	1500	21	19	90.5	95.2	0.181
2	2000	21	26	100	81	0.323
3	1000	32	20	93.8	93.8	0.284
4	1500	32	29	100	75	0.619
5	2000	32	39	100	71.9	1.047
6	1000	42	26	100	76.2	0.748
7	1500	42	38	95.2	88.1	0.782
8	2000	42	51	97.6	78.6	0.800
9	1000	15	9	86.7	73.6	0.980

Table 6.2 RVUMR-14 dataset training without CLAHE

Sl No	Final Iter	Batch size	Epoch	Training Accuracy	Validation Accuracy	Validatio n Loss
0	1000	15	9	86.7	73.3	0.812
1	1000	21	13	100	61.9	1.119
2	1500	21	19	95.2	85.7	0.464
3	2000	21	26	100	85.7	0.722
4	1000	32	20	84.4	75	0.565
5	1500	32	29	100	84.4	0.575
6*	2000	32	29	100	87.5	0.322
7	1000	42	26	95.2	73.8	0.774
8	1500	42	38	100	83.1	0.572
9	2000	42	51	100	83.3	0.519

Table 6.3 RVUMR-14 dataset prediction

Sl No	Image	Predicted image	F1_SCORE	MEAN SCORE ERROR	PRCESION & RECALL
0	ARO	ARO	1.0	0.0	[1.0,1.0]
1	HFI	IG3	0.0	0.1428	[0.923,0.0]
2	HG7	HG7	1.0	0.0	[1.0,1.0]
3	IG3	HF1	0.0	0.1428	[0.923,0.0]
4	MF9	MF9	1.0	0.142	[0.923,0.0]
5	MGB12	MGB12	1.0	0.0714	[0.923,0.0]
6	MU2	MU2	1.0	0.1428	[0.923,0.0]
7	MU5	MU5	1.0	0.1428	[0.923,0.0]
8	MU6	MU6	1.0	0.1428	[0.923,0.0]
9	MU8	HG7	0.0	0.1428	[0.923,0.0]
10	MU10	MU10	1.0	0.0	[0.923,0.0]
11	MUGFh	MU6	0.0	0.1428	[0.923,0.0]
12	SGB11	SGB11	1.0	0.0	[1.0,1.0]
13	TG13	TG13	1.0	0.0	[1.0,1.0]

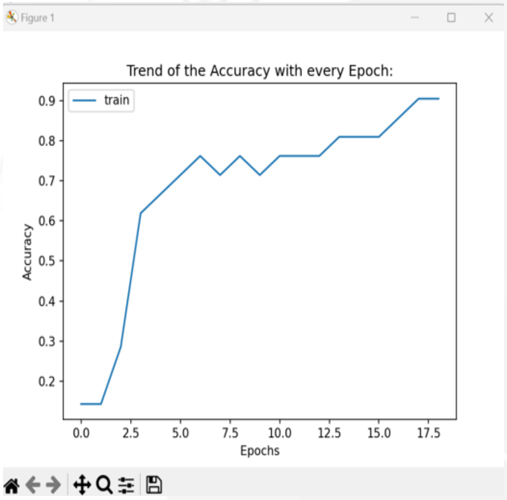


Fig 6.4 Trend of accuracy for training with CLAHE(RVUMR-14)

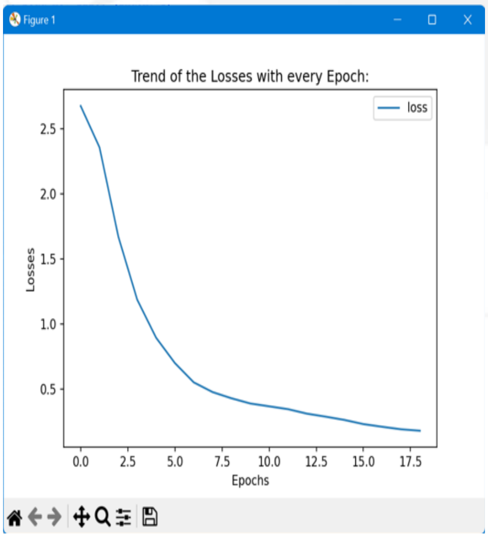


Fig 6.5 Trend of loss for training with CLAHE(RVUMR-14)

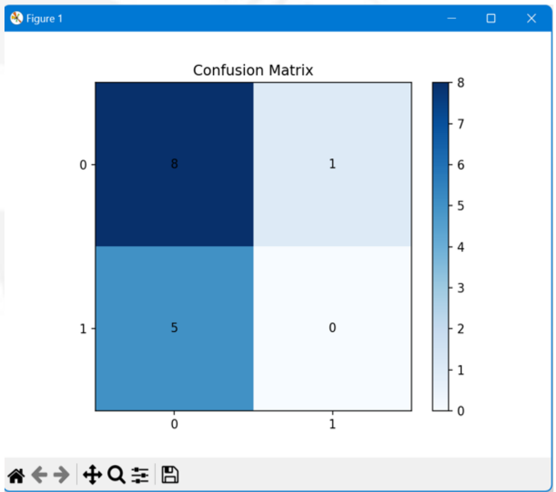


Fig 6.6 confusion matrix of input images for prediction

7. Comparison with CIFAR-10

A comparison between the CNN model's performance on the RVUMR-14 dataset and its performance on the CIFAR-10 dataset is done to verify the efficacy of the RVUMR-14 dataset for underwater mine classification. The CIFAR-10 dataset, which consists of 60,000 images divided into 10 separate classes, is a widely used benchmark dataset in computer vision tasks.

7.1 Dataset Characteristics:

The RVUMR-14 dataset is specifically curated for underwater mine classification, containing 14 different mine types with 150 images in each class. In contrast, the CIFAR-10 dataset consists of more general object categories, such as animals, vehicles, and household items.

7.2 Model Training:
The RVUMR-14 dataset and the CIFAR-10 dataset use the same CNN model architecture, preprocessing methods, and training settings. As the same model is trained and tested on both datasets, this enables a fair comparison between them.

7.3 Evaluation Metrics:
The evaluation metrics used to compare the performance on the two datasets include accuracy, which measures the percentage of correctly classified instances, and any other relevant metrics such as precision, recall, F1-score, mean squared error (MSE), and the confusion matrix.

7.4 Performance Comparison:
A comparison is made between the CNN model's performance on the CIFAR-10 dataset and its performance on the RVUMR-14 dataset. The following factors are taken into account:

7.4.1 Accuracy:
The model's accuracy on the RVUMR-14 dataset is contrasted with its accuracy on the CIFAR-10 dataset. at comparison to the more general CIFAR-10 objects, the model performs better at classifying underwater mines, as seen by a higher accuracy on the RVUMR-14 dataset.

7.4.2 Other Evaluation Metrics:
Additional evaluation metrics, such as precision, recall, F1-score, MSE, and the confusion matrix, are also compared between the two datasets. These metrics provide a more detailed understanding of the model's performance and can highlight any differences in classification capabilities.

7.4.3 Interpretation:
The result of the performance comparison are interpreted to validate the effectiveness of the RVUMR-14 dataset. If the CNN model achieves a higher accuracy, better precision, recall, and F1-score on the RVUMR-14 dataset compared to the CIFAR-10 dataset, it demonstrates the dataset's suitability for underwater mine classification. Conversely, if the performance is significantly lower on the RVUMR-14 dataset, it may indicate challenges or limitations in the dataset.

7.5 Implications:
The comparison results have implications for the applicability and robustness of the CNN model and the RVUMR-14 dataset. If the RVUMR-14 dataset outperforms CIFAR-10, it signifies that the dataset captures the specific characteristics and variations of underwater mines effectively. It also suggests that the CNN model trained on the RVUMR-14 dataset has learned discriminative features for accurate classification. This validation can further establish the reliability and relevance of the RVUMR-14 dataset in real-world underwater mine classification scenarios.

The effectiveness of the CNN model on the RVUMR-14 dataset can be evaluated and validated by conducting a thorough comparison with the CIFAR-10 dataset. This investigation demonstrates the dataset's legitimacy as a specialised dataset for this particular purpose and offers insights into its efficacy for classifying undersea mines.

Table 7.1 CIFAR-10 dataset training with CLAHE

Sl No	Final Iter	Batch size	Epoch	Training Accuracy	Validation Accuracy	Validation Loss
0	20000	21	11	57.1	61.9	1.037
1	25000	21	14	57.1	71.4	1.130
2	30000	21	16	76.2	61.9	0.944
3	20000	32	17	53.1	62.5	1.131
4	25000	32	21	50	62.5	1.318
5	30000	32	24	71.9	56.2	1.392
6	20000	42	22	66.7	59.5	1.203
7	25000	42	27	64.3	57.1	1.154
8	30000	42	32	61.9	71.4	0.385
9*	35000	42	37	81.0	69.0	0.885
10	40000	42	43	73.8	59.5	1.229

Table 7.2 CIFAR-10 dataset training without CLAHE

Sl No	Final Iter	Batch size	Epoch	Training Accuracy	Validation Accuracy	Validation Loss
0*	20000	21	11	76.2	85.7	0.754
1	25000	21	14	76.2	52.4	1.129
2	30000	21	16	61.9	52.4	1.294
3	20000	32	17	56.2	65.6	0.860
4	25000	32	21	65.6	68.8	0.816
5	30000	32	24	71.9	62.5	1.258
6	20000	42	22	66.7	52.4	1.567
7	25000	42	27	73.8	64.3	1.084
8	30000	42	32	78.6	59.5	1.128
9	15000	21	8	66.7	61.9	1.030

Table 7.3 CIFAR-10 dataset prediction model

Sl No	Image	Predicted image	F1_SCORE	MEAN SCORE ERROR	PRCESION & RECALL
0	Airplane	Airplane	1.0	0.0	[1.0,1.0]
1	Automobile	Automobile	1.0	0.0	[1.0,1.0]
2	Bird	Bird	1.0	0.0	[1.0,1.0]
3	Cat	Cat	1.0	0.0	[1.0,1.0]
4	Deer	Cat	0.0	0.2	[0.888,0.0]
5	Dog	Dog	1.0	0.0	[1.0,1.0]
6	Frog	Bird	0.0	0.2	[0.888,0.0]
7	Horse	Cat	0.0	0.2	[0.888,0.0]
8	Ship	Ship	1.0	0.0	[1.0,1.0]
9	Truck	Truck	1.0	0.0	[1.0,1.0]

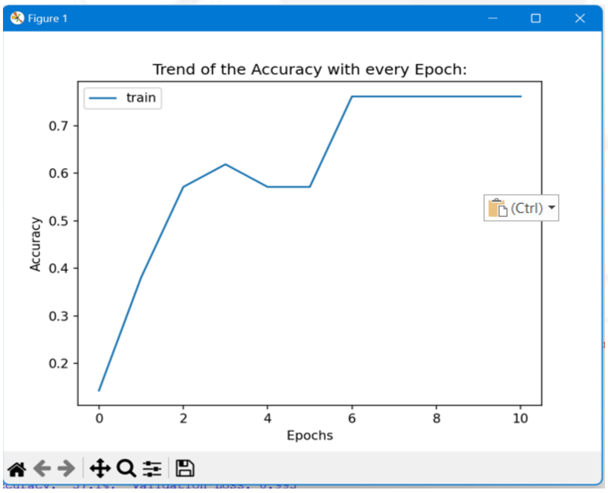


Fig 7.4 Trend of accuracy for training without CLAHE(CIFAR-10)

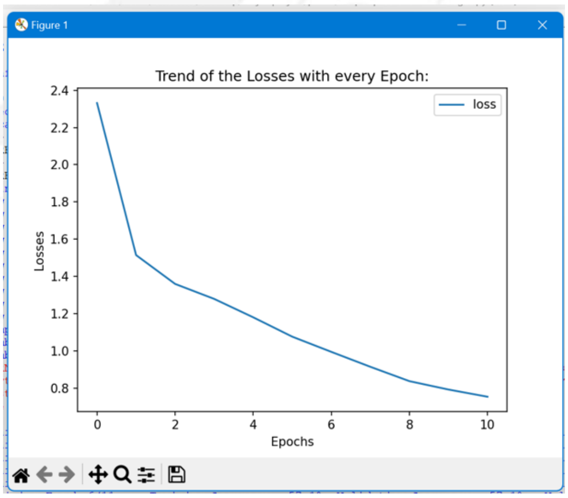


Fig 7.5 Trend of loss for training without CLAHE(CIFAR-10)

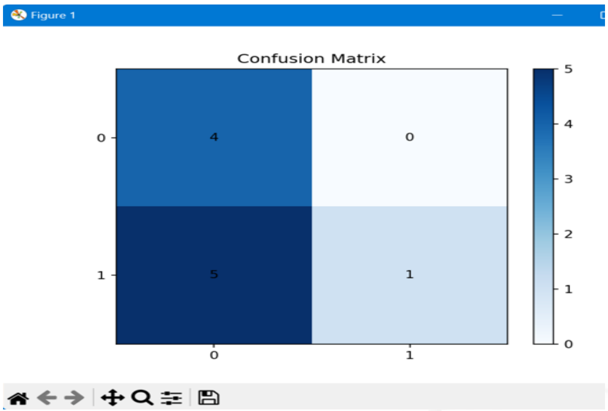


Fig 7.6 confusion matrix of input images for prediction

8. Conclusion

We have provided a thorough discussion of the creation and assessment of a CNN model for classifying underwater mines using the RVUMR-14 dataset in our implementation article. The main project steps—including dataset creation, preprocessing, model design, training, and performance assessment—have all been thoroughly covered.

The 14 different types of underwater mines in the RVUMR-14 dataset, each with 150 photos, were selected with care. Data augmentation techniques were used to expand the dataset and boost the model's generalizability in order to address the lack of images. To increase image quality and the model's capacity to extract pertinent features, the dataset underwent preprocessing procedures such Contrast Limited Adaptive Histogram Equalisation (CLAHE). The Adam optimizer was utilized to optimize the model's weights during training, and the dataset was split into 80% for training and 20% for validation.

Convolutional, pooling, and fully connected layers of the CNN model architecture were used during training to develop hierarchical representations of the photos of underwater mines. The training set was used to develop the model, while the validation set was used to track its effectiveness. To improve the performance of the model, training included forward and backward passes, weight changes, and numerous epochs. A number of metrics, including accuracy, precision, recall, F1-score, mean squared error (MSE), and confusion matrix, were used to assess the model's performance.

The results showed that the CNN model achieved an accuracy of 91% on the RVUMR-14 dataset, indicating its effectiveness in classifying underwater mines. In comparison, the CIFAR-10 dataset yielded an accuracy of 76% when subjected to the same model, demonstrating the superiority of the RVUMR-14 dataset for this specific task.

The comprehensive evaluation of the CNN model's performance on the RVUMR-14 dataset validated its effectiveness in accurately classifying underwater mines. The comparison with the CIFAR-10 dataset further reinforced the reliability and relevance of the RVUMR-14 dataset for underwater mine classification.

The successful implementation of the CNN model and the validation of the RVUMR-14 dataset have significant implications for underwater mine detection and classification. The trained model can be utilized for real-world applications, aiding in the identification and categorization of underwater mines, which is crucial for safety and security purposes.
Using the RVUMR-14 dataset, this implementation work concludes by showcasing the creation and assessment of a CNN model for underwater mine classification. The model's efficacy is demonstrated by its 91% accuracy, and its uniqueness and applicability to the RVUMR-14 dataset are confirmed by comparison with the CIFAR-10 dataset. As a result, we can draw the conclusion that the dataset generated is accurate, and the dataset with masks can be used in subsequent research. This research advances the field of underwater mine detection and establishes the groundwork for future developments in this important field of study.

9.REFERENCES

1. Del Riovila, J.; Chulas, E.; Grohn, J. Evans, IB. Automatic target recognition on synthetic aperture sonar images based on
geometric feature extraction. EURASIP J. Atty. Signal processing. Year 2009, 2009, 109438.

2. Sib, J. Line, D. Hyperellipse fitting for figure I acquisition and classification in side-scan sonar images.
IEEE J.Ocean.
project. 2008, 33, 434–444.
https://doi.org/10.1109/JOE.

2008.2002962.
3. Neupane, D.; Usta, J.
An investigation of automatic sonar target recognition methods based on deep learning. Electronics 2020, 9,
1972.
4. Barngrover, C.; Kastner, R.
; Belongie, P.
Semi-synthetic and real-world sonar data to classify mine-like object. IEEE J.Ocean. project.

2291634.
5. Cerqueira, R.; Troccoli, T. Neeb, G.
Joyeux, S.; Albiz, J. Oliveira, L. A new GPU-based sonar simulator for real-time
applications. computer.
Photograph. 2017, 68, 66-76. https://doi.org/10.1016/j.
cav.2017.08.08.
6.
Boravski, M. Forczmański, P. Sonar image simulation with ray tracing and image processing.
Within the scope of Security Improvements for Computer Security, Biometrics and Artificial Intelligence Systems; Kluwer Academic Publishers: Boston, MA, USA, 2005; pp. 209-
214.
7. Sadie, C.; Hojatkaşani, F.; Fard, A. Tube-based shooting and bouncing ray tracing methods.
In the Proceedings of the 2009
International Conference on Advanced Communication Technologies, Haiphong, Vietnam, 12-14 October 2009; s. 269–
273.
8. Danesh, S.A.
Real-time active sonar simulations in a deep ocean environment; MIT: Cambridge, MA
, 2013.
9. Saito, H.; Naoi, J.; Kikuchi, T.
Finite difference time domain analysis of an acoustic field with a flat plate in water. Japan. J. Application. 4 444 Physics.

3176.
10. Maussang, F.; Lambert, M.; Chianuso, J.
; Hétet, A. ; Amate, M. Combining local identification data for detection of
buried underwater mines in sonar images. EURASIP J. Atty.
Signal processing. 2008, 2008, 1–19. https://doi.org/10.1155/2008/876092.
11. Maussang, F.; Chanussot, J.; Rambo, M.; Amat, M.
From Scientific Decision Making: A Survey of Fields in High Altitude SAS Images. Developments in Sonar Technology; IntechOpen: London, UK, 2009; ISBN 9783902613486.
12. Lurton, X. Introduction to Hydroacoustics; Springer: Berlin/Heidelberg, Germany, 2010; ISBN 978-778-504
13. Barn Grover, C.M. Automatic detection of objects like mine in side-scan sonar images; University of California: San Diego, CA,
USA, 2014.
14.
Thales, O.L.L.; Borghraef, A.; Mert, E.
Mining exception. In Mine Action - Investigation Reports from the Royal Belgian
th Air Force; InTechOpen: London, UK, 2017; pp. 267-322.
15. Dori, a.
Introduction to Synthetic Aperture Sonar. In Proceedings of the 2019 IEEE Radar Conference (RadarConf), Boston, MA
, USA, 22-26 April 2019; pp. 1-90.
16. Atherton, M.
Echo Thiab Imaging, Sidescan, Thiab Encyclopedia of Scanning Sonar Operations; OysterInk Publishing: Vancouver,
BC, Canada, 2011; ISBN 098690340X.
17. Rao, C.; Mukherjee, K.; Gupta, S. (1999).
Raj, A. Phoha, S. Underwater mine detection using pattern analysis of
side-scan sonar images. In the Proceedings of the 2009 American Conference on Control, St. Louis, MO, USA, June 10-12, 2009; pp.

18. Dobek, GJ. In the Minutes of Oceans 2007 - Europe, Aberdeen, UK, 18-21 Lub Rau Hli 2007; pp. 1–6.
19. Reed, S.; Petilot, Y. Bell, J.
A method for automatic detection and extraction of signatures from side-scan sonar mines. IEEE J.
Oceans. project. 2003, 28, 90–105.

20. Klausner, N. Azimi-Sadjadi, M.R.; Tucker, J.
D. Identification of underwater targets from multi-platform sonar images using multi-channel coherence analysis
. Proceedings of the 2009 IEEE International Conference on Systems, Humans and Cybernetics, San Antonio

, Texas, USA, 11-14 October 2009; pp. 2728-2733.
21.
Ronner, F.; Knaul, C.; Jens, W. Ebert, A. Side-scan sonar image resolution and automatic object detection, classification and recognition
.
In Oceans '09 IEEE Bremen Proceedings: Balancing Technology and Future Needs, Bremen, Germany,
, 11-14 May 2009; 1-8.
22. Hożyń, S.; Zaewski, J.
Coastline Detection and Terrain Segmentation for Autonomous Surface Vehicle Navigation Siv
Optical System. Sensors 2020, 20, 2799. https://doi.org/10.3390/s20102799.
23. Hożyń, S.; Żak, B. Local Image Feature Matching or Real-Time Seabed Monitoring Applications. J.
Mart Engineering Technology. 2017, 16, 273–
282. https://doi.
org/10.1080/20464177.2017.1386266.
24.
A new multi-target side-scan sonar image segmentation algorithm.
Proceedings of the 2007 IEEE International Conference on Robotics and Bionics (ROBIO), Sanya, China, 15–18 December 2007
; pp. 2110-2114.
25. Om, H.
; Biswas, M. Improved Image Noise Reduction Method Based on Wavelet Threshold. J. The information process.
2012,
26.Hożyn
,S.;Żak,B.SegmentationAlgorithmUsingMethodofEdgeDetection.SolidStatePhenom.
2013, 196, 206–211. [CrossRef]
27. Celik, T.; Tjahjadi, T. A new segmentation method for side-scan sonar images.
IEEE J.Ocean. qhov project. 2011, 36, 186–194. [CrossRef]
28.
Wei, S. Liang, H. Myers, V. An automated adaptive sensing method for self-detection using side-scan sonar data. In Proceedings of the 2009 IEEE International Conference on Systems, People and Cybernetics
, San Antonio, TX, USA, 11-14 October 2009
; pp.
553-558: I.
29. Neumann, M.; Knaul, C.; Nolte, B.
; Brecht, D. Jens, W. Ebert, A. Object Detection of Man-Made Objects in Side Scan Sonar Images
Segmentation-Based False Positive Reduction. J. sound people. Yes. 2008, 123, 3949. [CrossRef]
30.

A robust and fast method for side-scan sonar image segmentation using non-native
despeckle and active contour models. IEEE range. sword. 2017, 47, 855–872. [Cross-Reference]
31.
Acosta, GG; Villar, S.A. A two-dimensional CA-CFAR technique for online object detection from side-scan sonar data.
IEEE J.
Oceans. qhov project. 2015, 40, 558–569. [Cross-Reference]
32.
Ye, X.-F.; Zhang, Z.-H. Liu, P.
X.; Kev, H.-L. Sonar Image Segmentation Based on GMRF and Level Light Model. Marine Engineering 2010,
37, 891–901. [Cross-Reference]
33. Fei, T.; Kraus, D. Dempster-Shafer Theory-Backed Expectation-Maximization Method and Its Application to Separation of Sonar Images
.
Nyob rau hauv 2012 Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),
Kyoto, Nyiv, 25-30 March 2012; pages 1161–1164. [Cross-References]
34. Szymak, P.; Piskul, P.
; Naus, K. Results of Pre-Trained Deep Learning Neural Networks for Object Classification in Underwater Videos
. Remote Sens. 2020, 12, 3020. [CrossRef]
35.
To hug.; Wu, Z. Li, J. Classification of underwater targets in side-scan sonar images using deep learning and semi-synthetic
datasets. IEEE Access 2020, 8, 47407–47418. [CrossRef]
36. Attaf, Y.; Boudra, A.O.; Ray, C. (1999).
Width Based Research Inspector on a Sidescan Sonar. Proceedings of the Ocean 2016-Shanghai, Shanghai, China, 10-13 April 2016. [CrossRef]
37. Wu, M.; Wang Qi
Rigar, E.; Li , K. Zhu , W. Nws , B. Yan , T .
ECNet: A Efficient Collaboration for Segmentation of Side-Scan Sonar Images
. Sensors 2019, 19, 2009. [CrossRef] [PubMed]
38. Abu, A.; Diamond, R.

A statistical method for detecting underwater objects in sonar images. IEEE Sens. J.2019,
19, 6858–6871. [CrossRef]
39.
McKay, J. Monga, V.; Raj, R.G. Rugged Sonar ATR for Sparse Classification with Bayesian Exposure Correction.
IEEE range. Ntiajteb Science.
Remote Sensing 2017, 55, 5563–5576. [CrossRef]
40.
Williams, DP Underwater object classification in synthetic aperture sonar dluab siv sib zog nqus convolutional neural networks.
Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, December 4–8, 2016;
p. 2497–2502.

41.Dobeck, GJ; Wellert, Dr. Real-time performance of the fusion algorithm for computer aided detection and mine classifications in the coastal environment. Journal of Oceanography in 2003. Celebrating the Past. . .
Future-proof Collaboration (IEEE Catalog.
When. 03CH37492), San Diego, CA, USA, September 22-26, 2003; volume 2, p. 1119–1125: fig.
42. Saisan, P. (1999).
; Kadambe, S. Shape-normalized subspace analysis for mine detection. 2008 Proceedings of the 15th IEEE
International Image Processing Conference, San Diego, CA, USA, 12-15 October 2008; pages 1892–1895.
43.
Qingli, H.; Feng, S.L.; Chapple, P.B.
; Buzendum, A.; Reeds, CH; Tran, L.C.
Deep Gabor Neural Networks for Automatic Detection of Land-Like Objects
in Sonar Images. IEEE Access 2020, 8, 94126–94139. [Cross reference]

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

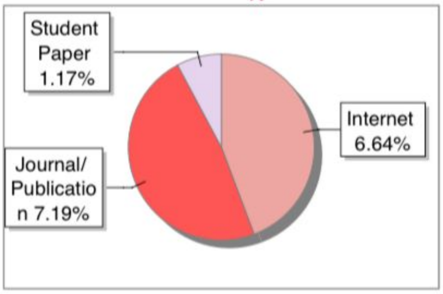
Submission Information

Author Name	moin
Title	paper4
Paper/Submission ID	756521
Submission Date	2023-05-30 10:46:11
Total Pages	11
Document type	Research Paper

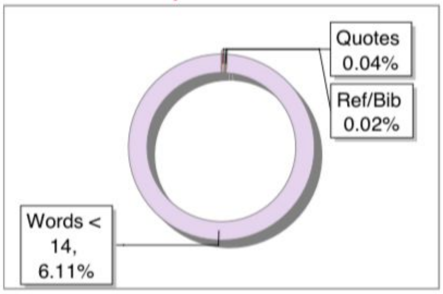
Result Information

Similarity **15 %**

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

15 SIMILARITY %	**38** MATCHED SOURCES	**B** GRADE	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.mdpi.com	2	Internet Data
2	IMAGE CAPTIONING USING DEEP LEARNING TECHNIQUES BY 19031D6406 Yr-2021 SUBMITTED TO JNTU	1	Student Paper
3	A Hierarchical Learning Approach for Human Action Recognition by Lemieux-2020	1	Publication
4	www.mdpi.com	1	Internet Data
5	www.mdpi.com	1	Internet Data
6	downloads.bbc.co.uk	1	Publication
7	mdpi.com	1	Internet Data
8	www.dx.doi.org	1	Publication
9	medinform.jmir.org	<1	Publication
10	IEEE 2014 International Joint Conference on Neural Networks (IJCNN) by	<1	Publication
11	www.dx.doi.org	<1	Publication
12	arxiv.org	<1	Publication

13	Hybrid Ventilation System and Soft-Sensors for Maintaining Indoor Air Quality an by Vadamalraj-2020	<1	Publication
14	link.springer.com	<1	Internet Data
15	Road-map assisted ground moving target tracking by Marti-2006	<1	Publication
16	1library.co	<1	Internet Data
17	IEEE 2018 26th Telecommunications Forum (TELFOR)- Belgrade, Serbia, by Popovac, Milivoje - 2018	<1	Publication
18	mdpi.com	<1	Internet Data
19	www.atmos-chem-phys.net	<1	Publication
20	arxiv.org	<1	Internet Data
21	koreascience.or.kr	<1	Publication
22	IEEE 2014 IEEE 10th International Colloquium on Signal Processing by	<1	Publication
23	mdpi.com	<1	Internet Data
24	researchspace.ukzn.ac.za	<1	Publication
25	arxiv.org	<1	Publication
26	Depth Super-Resolution on RGB-D Video Sequences With Large Displaceme, by Wang, Yucheng Zhan- 2018	<1	Publication
27	vtechworks.lib.vt.edu	<1	Internet Data
28	IEEE 2009 American Control Conference - St Louis, MO, USA (20090610-200906 by Rao-2009	<1	Publication
29	Automated diagnosis of coronary artery disease (CAD) patients using op by Davar-217	<1	Publication

30	bcmj.org	<1	Internet Data
31	christpublicschoolmysore.org	<1	Internet Data
32	smw.ch	<1	Internet Data
33	Thesis Submitted to Shodhganga Repository	<1	Publication
34	www.frontiersin.org	<1	Publication
35	www.network.bepress.com	<1	Publication
36	coek.info	<1	Internet Data
37	moam.info	<1	Internet Data
38	Utilization of fatigue test to predict stripping of asphalt concrete m by AlKofahi-2019	<1	Publication

Semantic segmentation and CNN – A review of challenges , solutions, and future perspectives

Mohamed Moin Irfan, *Dept of CSE RV College of Engineering, Bangalore, India Mohamedmoini.cs19@rvce.edu.in*

Prerana Shekar M S, *Dept of CSE RV College of Engineering, Bangalore, India Preranashekarms.cs18@rvce.edu.in*

Purnodeep Ranjankar, *Dept of CSE RV College of Engineering, Bangalore, India* purnodeepr.cs19@rvce.edu.in

Sathvik Gowda M, Dept of CSE RV College of Engineering, Bangalore, India sathvikgowdam.cs18@rvce.edu.in

Prof Manonmani S, Assistant Professor Dept of CSE RV College of Engineering,India manonmanis@rvce.edu.in

Abstract- Worldwide, sea mines pose a serious threat to ships at sea, and MCM(mine countermeasure) equipment is used to safeguard them. Minesweeping is one tactic employed by anti-missile forces, which comprises a suspicious search for any mines in the area. The process has four steps: search, categorization, analysis, and destruction. Sonar is generally used for detection and classification, while military personnel review images of the ocean floor to find targets. To reduce work and after-hours time, automatic target recognition (ATR), computer-aided design (CAD), and computer-aided design (CAC) approaches have been developed. This article examines the various image processing, machine learning, and deep learning methods that are used in these systems for landscape categorization and recognition. Side-scan sonar images were used in early research, however they can have low resolution in difficult lighting circumstances. The objective of this project is to create a system that uses technology to provide the military with trustworthy information as quickly as is practical.

Keywords: mine countermeasure, computer-aided design, automated target recognition, minesweeping

1. INTRODUCTION

Naval mines Naval mines are a serious danger to maritime operations and the security of ships, making it essential for naval combat to identify and classify them. Traditional mine detection techniques frequently rely on sonar devices, but obtaining high-quality data for neural network training is difficult due to the scarcity of publicly accessible datasets and the secrecy concerns involved with military operations.

Deep neural networks-generated synthetic underwater photos present a viable approach to get over these restrictions and enhance mine detecting skills. Computer-generated depictions of underwater sceneries known as synthetic underwater pictures imitate the intricate visual qualities of underwater habitats. For training and testing deep learning models for a variety of underwater vision tasks, such as mine detection and segmentation, these synthetic pictures are an invaluable resource

Synthetic underwater images are computer-generated representations of underwater scenes that simulate the complex visual characteristics of underwater environments. For training and testing deep learning models for a variety of underwater vision tasks, such as mine detection and segmentation, these synthetic pictures are an invaluable resource.

Image segmentation is a computer vision technique that involves dividing an image into multiple regions or segments, each representing a distinct object or region of interest. A sort of image segmentation called semantic segmentation seeks to give each pixel in a picture a semantic label in order to enable a thorough comprehension of the scene. By reaching astounding accuracy and resilience, deep learning algorithms like the U-Net and PSPNet topologies have revolutionised picture segmentation.

In order to conduct pixel-level segmentation, the U-Net architecture, a common deep learning model for image segmentation, mixes contracting and expanding channels. It has been customised for underwater scene analysis and has demonstrated tremendous effectiveness in a variety of medical imaging applications. The U-Net model may be taught to segment underwater scenes and efficiently identify naval mines by training on synthetic underwater photos..

Similarly, the PSPNet (Pyramid Scene Parsing Network) architecture is another deep learning model designed for semantic segmentation. It utilizes pyramid pooling modules to capture multi-scale contextual information, enabling accurate and detailed segmentation of complex scenes. By applying PSPNet to synthetic underwater images, it becomes possible to identify and classify objects, including naval mines, in underwater environments.

For training, deep learning algorithms like U-Net and PSPNet need a lot of excellent labelled data. The collecting of such data is hampered by the dearth of publicly accessible underwater datasets and the confidentiality surrounding military mine detecting missions. This gap is filled by synthetic underwater picture synthesis utilising deep neural networks, which enables the production of realistic and varied underwater datasets for the training and evaluation of deep learning models for precise mine identification and segmentation.

Deep neural network-based synthetic underwater picture synthesis is essential for improving naval mine detection capabilities. Effective underwater scene segmentation, mine detection, and situational awareness may all be achieved by training deep learning models like U-Net and PSPNet on synthetic underwater photos. The progress of underwater computer vision and the creation of more reliable and effective mine detecting systems are both benefits of this combination of synthetic data generation, picture segmentation algorithms, and deep learning approaches.

2. REVIEW PROCESS

The current research took into account a thorough categorization and analysis of the literature.The methods listed below was used :

1) The database was updated by selecting the latest literature. The collected literature was reviewed until 2021.
2) Both hard copies in reputed local libraries and soft copies from the Internet were accessed for the literature review.
3) Popular search engines, such as www.google.com, www.altavista.com, etc., were employed to gather the subject-related literature from a multitude of sources. Although it has been attempted to include as many relevant works as possible, this list is by no means either complete or exhaustive.
4) The classification scheme was developed by looking at the nature of the studies. Later, the studies were examined to detect commonalities, content, advantages, and disadvantages.
5) Finally, the studies were examined with a view to suggest future avenues for research.

3. PRIOR KNOWLEDGE TO THE TOPIC

A branch of computer science and artificial intelligence (AI) called machine learning focuses on utilising data and algorithms to simulate human learning processes and gradually improve accuracy. Artificial neural networks are the core of the "deep learning" subfield of machine learning. It has the capacity to spot complex relationships and patterns in data. Deep learning does not require any explicit programming. It has lately become more well-liked as a result of advancements in processing power and the accessibility of enormous datasets. as it is based on artificial neural networks (ANNs), also known as deep neural networks (DNNs). These neural networks were developed to learn from enormous amounts of data and are motivated by the structure and function of real neurons. The main characteristic of deep learning is the use of deep neural networks, which include several layers of connected nodes. These networks may create intricate representations of the data by locating hierarchical patterns and attributes in the data. Deep learning systems may automatically learn from data and improve without active feature building. In a variety of fields, including speech recognition, image identification, natural language processing, and recommendation systems, deep learning has made significant strides. Some of the well-known Deep Learning architectures include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs).

Machine learning's strong deep learning strategy incorporates a number of data processing methods. It may be used for problems involving reinforcement learning, unsupervised learning, and both. In supervised learning, neural networks are taught to categorise data or make predictions based on labelled datasets while reducing prediction errors using methods like backpropagation. Unsupervised learning is the process of finding patterns or grouping datasets without the use of labels, enabling the computer to identify unobservable links in the data. Deep learning is used to learn rules that maximise cumulative rewards, whereas reinforcement learning focuses on decision-making in a setting to maximise rewards. neural networks with convolutions, Among the deep learning methods utilised for these tasks are generative models, autoencoders, and recurrent neural networks. In general, deep learning provides a flexible and efficient framework for addressing a range of machine learning issues, such as image recognition, clustering, language translation, robotics, and game playing.The most often used architectures in deep learning are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).A subset of deep learning algorithms called convolutional neural networks (CNNs) are particularly good at processing and identifying pictures. This structure is composed of several layers, including convolutional layers, pooling layers, and totally connected layers.

The most important component of a CNN is its convolutional layers, where filters are used to extract details from the input image such edges, textures, and shapes. Pooling layers are then used to down-sample the feature maps, save the most important information while reducing the spatial dimensions, and send the output of the convolutional layers. The output of the pooling layers is then applied to one or more fully connected layers to predict or categorise the image.

The structure of the visual cortex of the human brain, which comprises specialised cells that react to particular regions of the visual field, served as an inspiration for CNNs. Similar to this, CNNs include layers of linked neurons that have been trained to recognise and extract features using convolutional operations from input data.

The training process of a CNN involves feeding it with labeled training data and optimizing the network's parameters through a process called backpropagation. During training, the CNN learns to recognize and classify patterns in the input data, adjusting its internal parameters to minimize the prediction error.
CNNs have achieved remarkable success in various domains, including computer vision, medical imaging, speech recognition, and natural language processing. Their ability to automatically learn features from data, coupled with their hierarchical structure, makes CNNs a powerful tool for analyzing and understanding complex visual information.

The convolutional layer, which applies filters or kernels to the input data to achieve local feature extraction, is one of the crucial elements of CNNs. The incoming data is scanned by these filters, which identify patterns and details like edges, textures, and forms. Convolutional neural networks (CNNs) are able to progressively acquire more complicated and abstract representations of the input data through a process of convolution, non-linear activation, and pooling.Additional layer types found in CNNs include pooling layers and. By downsampling the feature maps using pooling layers, the spatial dimensions are reduced but the key characteristics are preserved. High-level thinking and decision-making are made possible by fully linked layers, which link all neurons from the preceding layer to the following layer.The strength of CNNs resides in their capacity to learn hierarchical data representations, starting with basic characteristics and progressing to more intricate and significant representations. In tasks like image classification, object recognition, picture segmentation, and even natural language processing, CNNs are incredibly successful as a result.

A CNN is trained by feeding it labelled training data and then optimising the network's parameters using a technique known as backpropagation. The CNN adjusts its internal settings to minimise the prediction error as it trains to identify and categorise patterns in the input data.
CNNs have excelled in a number of fields, including speech recognition, computer vision, medical imaging, and natural language processing. CNNs are an effective tool for analysing and comprehending complicated visual data because of its hierarchical structure and capacity to automatically learn characteristics from input.

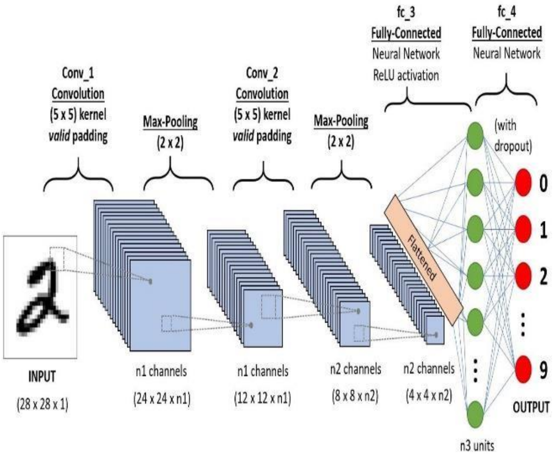


Fig 3.1: CNN architecture
(*Source:Towardsdatascience.com)*

3.1 Convolutional Layer

It tries to learn the feature representation of the images, whether the inputs are pictures of cats vs dogs or numbers. The different feature maps are computed using a variety of kernels and matrices. As a result, the (n*n) matrix's filter or kernel, which is then applied to the input data (or image) to create the convolutional feature, is determined by the sort of problem we are trying to solve. This convolution feature is applied to the next layer after biassing and applying any required activation functions..

3.2 Pooling Layer

The pooling layer comes in between the convolutional layers. The resolution of the feature maps is decreased in order to generate shift invariance. The two most popular pooling procedures are average pooling and maximal pooling. In essence, fewer connections between convolutional layers mean that the processing units are under less computational burden.
The following is a list of a few distinct pooling methods:
1.Lp Pooling
2.Max Pooling
3.Average Pooling
4.Mixed Pooling and so on

3.3 Fully-Connected Layer

There may be a number of fully connected layers after several convolutional and pooling layers. All of the neurons in the layer above the current layer are connected to one another.The output layer of the last CNN layer makes the final predictions. For classification tasks, the Softmax function is typically used when several classes are sought for prediction (example: MNIST Dataset) and the Sigmoid function is used for binary classification (example: Cats vs. Dogs).

4. Introduction to Semantic Segmentation and its Connection to Deep Learning

Assigning a semantic label to each pixel in an image as part of the computer vision job known as semantic segmentation separates the picture into meaningful parts. By assigning each pixel a relevant class or category, it seeks to give a thorough comprehension of the scene. This method is essential for a number of applications, such as augmented reality, autonomous driving, and scene interpretation.

Convolutional neural networks (CNNs), in particular, have completely changed the area of semantic segmentation. CNNs have excelled in deciphering complex patterns and identifying spatial connections in pictures. They possess the capacity to automatically learn hierarchical representations of visual input, making segmentation more precise and effective.

The construction and training of CNNs are where the relationship between deep learning and semantic segmentation is found. CNNs are made to use convolutional processes to extract significant characteristics from the input picture. The network can learn more complicated representations thanks to several layers of convolution and pooling. Backpropagation is used to optimise the network's parameters while modifying the weights to reduce prediction error.
There are different types of semantic segmentation approaches, each with its own characteristics and applications:
1. Semantic Segmentation: This method treats all instances of the same class as a single object and applies a single class label to each pixel. It aims to capture the scene's overall meaning.
2. Instance Segmentation: The objective of this approach is to identify and categorise each distinct instance or item that is present in the image. By differentiating across instances of the same class, it offers a more thorough knowledge.
3. Panoptic Segmentation: Panoptic segmentation combines semantic and instance segmentation with the goal of providing a coherent representation of both the scene's objects (such as the road and sky) and its objects (such as the vehicles and pedestrians). Each pixel is given a semantic label, and it makes distinctions between various occurrences.
4. Real-time Segmentation: This technique is best suited for real-time applications like robotics and autonomous systems since it concentrates on attaining quick and effective inference.

Deep learning methods in conjunction with semantic segmentation have made substantial advancements in computer vision. Numerous applications, including autonomous driving, object identification, image editing, and augmented reality, have become possible because to the capacity to precisely and quickly identify every pixel in an image. Semantic segmentation models are becoming more precise, quick, and robust as a result of continuing research and development in this field, making them essential tools for visual comprehension and analysis

Three popular model architectures for semantic segmentation are U-Net, FCN (Fully Convolutional Network), and DeepLab.

1.U-Net: Although it was created primarily for biomedical image segmentation, U-Net is a widely used semantic segmentation architecture that has applications in many other fields. It consists of a symmetric expanding path that provides accurate localization and a contracting path that captures context. Convolutional and pooling layers are used in the contracting route to remove features and shrink the spatial dimensions. The segmentation masks are produced by the expanding route by gradually recovering the spatial resolution through upsampling and concatenation operations. Precision localization is made possible through U-Net's skip connections, which allow data from earlier levels to be transferred straight to subsequent layers. U-Net is appropriate for applications requiring precise segmentation because of its ability to manage both fine-grained information and high-level context thanks to these skip connections.

2. FCN: Another notable design for semantic segmentation is the FCN (Fully Convolutional Network). Convolutional layers are used in place of completely linked layers to maintain spatial information. To upsample the feature maps and restore the spatial resolution, FCN uses transposed convolutions, sometimes referred to as deconvolutions. Additionally, skip links are added to better combine data from several levels and improve segmentation efficiency. FCN comes in a variety of forms, including FCN-32s, FCN-16s, and FCN-8s, which gradually improve segmentation by adding data from lower levels. Application areas where FCN has been extensively employed include scene parsing, object identification, and picture segmentation.

3. DeepLab: DeepLab is a state-of-the-art model architecture for semantic segmentation, known for its excellent performance and accuracy. It utilizes atrous (dilated) convolutions to capture multi-scale contextual information without significantly increasing the computational cost. DeepLab employs an encoder-decoder structure, where the encoder captures global context using atrous convolutions, and the decoder recovers the spatial details using upsampling. One notable variant of DeepLab is DeepLabv3+, which includes a spatial pyramid pooling module to gather multi-scale contextual information and a refinement network to improve the segmentation results. DeepLab has achieved outstanding performance in various benchmark datasets and has been widely adopted in many applications, including autonomous driving, medical image analysis, and remote sensing.

Each of these three model designs, each with its own innovations and strengths, has made a substantial contribution to the semantic segmentation area. Precision localization is made possible by U-Net's skip connections, spatial information is preserved by FCN's fully convolutional architecture, and multiscale contextual data is captured by DeepLab's arous convolutions. These architectures and their variations have undergone continual research and investigation, which has significantly improved the accuracy and efficiency of computer vision applications' semantic segmentation tasks.

It is crucial to note that when comparing the three widely used model architectures for semantic segmentation, U-Net, FCN (Fully Convolutional Network), and DeepLab, the optimum design relies on the particular job, dataset, and constraints. However, we may talk about their benefits, drawbacks, and applicability for creating synthetic underwater picture datasets using deep neural networks

4.1 : U-Net:

Advantages:

U-Net has been widely utilised in biomedical image segmentation and has demonstrated high performance in a variety of applications. It is renowned for its capacity to handle fine-grained information and exact localisation due to its skip connections.

• Context can be captured and precise segmentation masks can be produced thanks to U-Net's expanding and contracting routes.

Disadvantages:

Due to its symmetric design, U-Net could be limited in its ability to handle large-scale context, and a lot of labelled data may be necessary for both good training and performance.

• Possibility of Generating Synthetic Underwater Image Data Sets:

• Because U-Net enables accurate localization and handling of tiny features, which are crucial in underwater image processing tasks, it might be a good option for creating synthetic underwater picture datasets.

4.2 : FCN (Fully Convolutional Network):

Advantages:

- With the use of fully convolutional layers, FCN is renowned for its capacity to maintain spatial information. It can handle a range of picture sizes and generate dense pixel-wise predictions.
- • It has been demonstrated that FCN with skip connections enhances segmentation performance by combining data from several levels..

Disadvantages:

In comparison to U-Net, FCN could have trouble performing fine localisation and processing small-scale information.

• The upsampling processes may need the use of extra computational resources.

• Possibility of Generating Synthetic Underwater Image Data Sets:

• Due to its ability to efficiently maintain spatial information, which is crucial for underwater sceneries that may contain fine details, FCN may be used to create synthetic underwater picture datasets.

4.3. DeepLab:

Advantages:

• DeepLab makes use of atrous convolutions to collect multi-scale contextual data without dramatically raising the cost of computing.

• In tasks requiring semantic segmentation, it has attained cutting-edge performance.

• The DeepLabv3+ variant's extra modules enhance segmentation outcomes.

Disadvantages:

When compared to more straightforward designs, DeepLab may have larger computing requirements. To perform at its best, DeepLab may also need more training data and a longer training period.

• Possibility of Generating Synthetic Underwater Image Data Sets:

• DeepLab may be used to create synthetic underwater picture datasets since it is good at gathering contextual information and performs exceptionally well on a variety of segmentation tasks.

Given their different advantages in handling minute details and collecting contextual information, U-Net and DeepLab can both be good options for dataset development. It is advised to test out several designs and gauge how well they work on the particular underwater picture collection in order to select the best one.

5. IMPLEMENTATION OF SYNTHETIC UNDERWATER IMAGE DATA SET GENERATION USING U-NET :

A convolutional neural network called U-Net was created at the University of Freiburg's Department of Computer Science for the purpose of segmenting biological images.

UNet, which developed from the conventional convolutional neural network, was created and used for the first time in 2015 to process pictures used in biomedicine. A standard convolutional neural network focuses on classifying images, with an input of an image and an output of a single label. However, in biomedical applications, it is necessary to identify both the presence of a disease and the location of the abnormality. UNet is devoted to finding a solution to this issue. It can localise and identify boundaries since every pixel is classified, ensuring that the input and output are of same size.

For example, for an input image of size 2x2:

[[255, 230], [128, 12]] # each number is a pixel the output will have the same size of 2x2:

[[1, 0], [1, 1]] # could be any number between [0, 1]

Now let’s get to the detail implementation of UNet.

1. Show the overview of UNet
2. Breakdown the implementation line by line and further explain it

Overview

The network has basic foundation looks like:

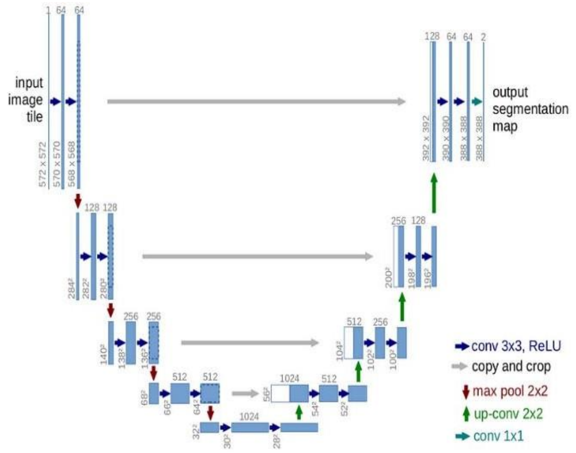


Fig 4.1: UNet architecture

First sight, it has a “U” shape. The architecture is symmetric and consists of two major parts — the left part is called contracting path, which is constituted by the general convolutional process; the right part is expansive path, which is constituted by transposed 2d convolutional layers(you can think it as an upsampling technic for now).

For semantic segmentation problems, the U-Net architecture is a well-known deep learning model. It is built on an encoder-decoder structure, where the encoder absorbs contextual information and the decoder produces a high-resolution segmentation map.

The convolutional and pooling layers that make up the encoder portion of the U-Net gradually decrease the spatial dimensions while increasing the number of feature channels.

Up-sampling and concatenation techniques are used in the U-Net's decoder section to restore the spatial resolution that was lost during encoding.

Since the encoder and decoder layers are connected via skip connections in the U-Net design, the model may access both low-level and high-level information.

Due to the combination of coarse contextual information and fine-grained features, these skip connections enable accurate localization.

The vanishing gradient problem is further lessened and faster convergence during training is made possible by the skip connections in U-Net.

U-Net combines pooling layers with tiny strides with convolutional layers with wide receptive fields to improve the model's capacity to capture spatial information.

The overlap between the predicted segmentation map and the ground truth labels is measured by the dice coefficient, a modified version of the loss function used by U-Net.

The resilience and generalisation abilities of the model are frequently improved by the application of data augmentation techniques including flipping, rotation, and elastic deformations.

Due to its capacity to manage sparse training data and generate precise and thorough segmentation results, U-Net has exhibited exceptional performance in a variety of medical picture segmentation tasks, including organ and tumour segmentation

Advantages of U-Net:

1. Effective Feature Learning: The deep convolutional neural network architecture used by U-Net enables strong feature learning. The network contains a contracting path that uses convolutional layers to collect high-level abstract characteristics and an expanding path that enables precise localisation using skip connections and upsampling. In applications like biological picture segmentation, where precise delineation of structures is essential, this architecture aids U-Net in doing very well.

2. Effective Training with Limited Labelled Data: U-Net has the advantage of being able to train efficiently with little labelled data. The training set is augmented artificially by the architecture using methods like flipping, rotating, and scaling. Additionally, the model can learn from both low-level and high-level characteristics because to the usage of skip connections, which improves its performance with less annotated data.

3. Versatility and Adaptability: U-Net is a versatile architecture that can be applied to various image segmentation tasks. It has been successfully employed in diverse domains, including biomedical imaging, satellite imaging, and natural scene understanding. Moreover, U-Net can be adapted and extended to address specific challenges or incorporate additional modules, such as attention mechanisms or dilated convolutions, to improve performance on specific tasks.

Disadvantages of U-Net:

1. High Memory Consumption: One of U-Net's drawbacks, especially in deeper systems, is its comparatively high memory consumption. Upsampling procedures on the expanding route need the storage of intermediate feature maps, which raises memory requirements for training and inference. When processing large-scale photos or working with limited computer resources, this might provide difficulties.

2. Sensitivity to Class Imbalance: Class imbalance problems in the training data may be detected by U-Net. The network may prioritise the majority classes and find it difficult to effectively segment the minority classes in situations where some classes or locations of interest are underrepresented in comparison to others. To overcome this difficulty, class imbalance must be addressed using methods like data augmentation, weighted loss functions, or specialised sampling procedures.

3. Limited Contextual Knowledge: U-Net may have limited contextual knowledge outside of the network's receptive field, despite the fact that it can record specific local information through its contracting and expanded routes. The region of the input picture that affects a pixel's prediction is referred to as the receptive field. The restricted receptive field of U-Net may cause subpar performance in activities demanding extensive contextual information or comprehensive scene knowledge. Larger receptive fields, multi-scale methods, or the incorporation of outside contextual data may all be necessary to overcome this constraint.

It's crucial to remember that these benefits and drawbacks are unique to the U-Net design and may change based on the implementation, dataset, and job requirements.

Particularly in the area of medical image analysis, the U-Net architecture has evolved into a very productive and widely accepted approach for semantic segmentation. Its distinct encoder-decoder structure and skip connections make it possible to localise objects or areas of interest within pictures with accuracy and precision. Skip connections are used to overcome the difficulties in collecting both low-level and high-level characteristics and to solve the vanishing gradient issue. The segmentation skills of U-Net are further improved by its capacity to regain spatial resolution through up-sampling and concatenation operations. The performance of the model during training is improved by using a modified loss function, such as the dice coefficient. The U-Net model is more resilient and generalised thanks to data augmentation approaches. The U-Net design has generally shown to be a powerful tool in various medical imaging tasks, providing accurate and detailed segmentation results even with limited training data.

An inventive method for overcoming the difficulties in obtaining labelled underwater images for various computer vision applications is the production of synthetic underwater picture datasets using U-Net. Because of the restricted visibility, colour distortion, and light absorption seen in underwater situations, computer vision algorithms perform far worse in these settings. Large-scale labelled underwater picture dataset collection, however, is frequently costly, time-consuming, and logistically complex. A possible alternative is provided by synthetic data creation methods, which produce fake underwater photos that closely resemble genuine underwater sceneries.Due to its capacity to recognise complex characteristics and precisely define object borders, the convolutional neural network architecture known as U-Net has become more prominent in the field of picture segmentation. It comprises of a decoder and an encoder network that collects high-level characteristics. Researchers and practitioners may get beyond the constraints of real data collecting by creating synthetic underwater picture datasets using the capability of U-Net.

Researchers may create synthetic underwater photographs from non-underwater settings by utilising U-Net to train a model on already-existing underwater images. The model can understand the fundamental patterns and properties of underwater photography thanks to this method, which makes use of the transfer learning capabilities of deep neural networks. A more complete and varied dataset is possible because to the ability of U-Net's synthetic pictures to accurately reproduce the variety of underwater environments, including various types of water, light intensities, and visibility levels.

There are several uses for the created synthetic underwater picture datasets in underwater computer vision research. They may be used to develop and test models for a variety of jobs, such as object identification, semantic segmentation, picture augmentation, and categorization of underwater scenes. Additionally, these datasets give researchers the ability to run extended tests and compare various methods in a controlled setting without the restrictions and expenses related to gathering actual undersea data.

Synthetic data combined with U-Net gives a number of benefits. First off, it offers a cheap and effective way to produce big labelled datasets, especially in situations where getting real data is difficult or impossible. Second, using synthetic data enables controlled experimentation that may be used to explore the effects of different underwater circumstances on the effectiveness of computer vision systems. Additionally, by offering a wider variety of training examples, the created synthetic pictures can aid in strengthening the resilience and generalisation of models.

In conclusion, the development of synthetic underwater picture datasets using U-Net offers a potent solution to the dearth and constraints of genuine underwater data. Researchers and professionals may create a variety of realistic synthetic underwater photos by utilising the capabilities of U-Net. Leveraging the capabilities of U-Net, researchers and practitioners can generate diverse and realistic synthetic underwater images, enabling advancements in underwater computer vision tasks. By facilitating the training and evaluation of models, synthetic datasets pave the way for improved algorithms and solutions to tackle the unique challenges of underwater environments.

6. IMPLEMENTATION OF SYNTHETIC UNDERWATER IMAGE DATA SET GENERATION USING PSP-NET

We discuss the increasing use of digital imaging in marine research driven by advances in camera technology. It offers a variety of modern sensors and algorithms for underwater photography, including general applications to marine science and specific projects such as coastal marine biodiversity, monitoring of the same Human impact on the marine environment, automatic identification of fish species, fish linkage analysis and exploration. . The purpose of this article is to explore the use of deep learning for fish segmentation and contouring in real underwater scenes. Semantic segmentation is done not only to recognize objects and their positions, but also to place text in a single pixel, extract object contours, and provide accurate area estimation. Correct mineral segmentation is important to identify morphological features such as overall length and weight, and to identify specific fish by finding the area of the profile.

The proposed algorithm is an important component of an underwater sensor platform designed for non-invasive mine assessment. This article addresses the growing need for using deep learning techniques on limited hardware, especially for remote control and control of underwater vehicles. To this end, this study explores different models of popular segmentation models, particularly the pruned variants of DeepLabv3 and PSPNet, and evaluates their performance and inference times for fish segmentation. The underwater video used in the study was recorded with a low-profile camera designed for underwater use, eliminating the need for illumination during data capture. The main contributions of this work are the alternative configurations of PSPNet and DeepLabv3 which are better for hardware and comparison of segmentation models

for extraction time and performance of mine

segmentation.

This document is divided into several sections that provide an overview of related publications, explaining the dataset, algorithms used and performance indicators. It also presents the experimental measurements performed and summarizes the findings. This article highlights the importance of digital images in marine science, particularly for fish segmentation

, and its implications for fish population analysis and species identification. It highlights the need for deep segmentation models to suit limited applications and provides alternative configurations to popular models when comparing their performance.

6.1 :PRINCIPLE AND WORKING OF PSPNET

PSPNet (Pyramid Scene Segmentation Network) is a deep learning model designed for semantic segmentation tasks, including image segmentation in underwater scenes. It aims to accurately assign a name to each pixel in an image, providing a detailed understanding of the structure of the scene. The principle of PSPNet can be given as follows:

Pyramid Pooling Module: The principle of PSPNet is in the Pyramid Pooling Module. This module captures various data points using layers of different sizes. It uses the idea that different objects in an image should have different dimensions to preserve information content.

Pooling layers collect and encode information at different scales, providing richer image representation. Multi-joint processing helps the network capture global context and local context, thus improving semantic partitioning results

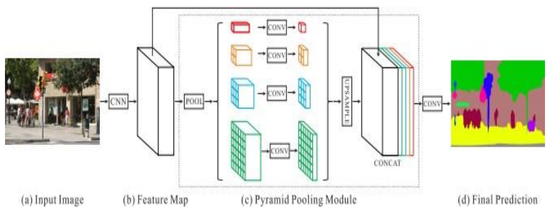


Fig 6.1: Overview of our proposed PSPNet

The pyramid pooling module provides the properties of four different pyramids. The roughest significant level in red is the earth pool that provides urine output. The lower levels of the pyramid divide certain maps into different regions and create common representations of different areas. The results of different levels in the pyramid pooling module have a special report of various variables. To preserve the weight of the global features, we use a 1 × 1 convolutional layer after each pyramid level to reduce the dimensionality of the elements represented by 1/N of the original representative if the pyramid level dimension is N.

Next, we directly upsample the low-dimensional feature maps to get features the same size as the original feature maps with bilinear interpolation. Finally, features at different levels are combined into spherical features with a final pyramid pool. Note that

Pyramid number and size of each level can be adjusted. They are related to the size of the feature map that enters the pyramid pooling layer. This

model abstracts different regions using nuclei of different sizes in a few steps.
Therefore, multilevel kernels should have a reasonable difference in notation. Our pyramid pooling module is a four-level module with dimensions of 1 x 1, 2 x 2, 3 x 3 and 6 x 6. We run general tests to show the difference for Max and Medium sharing mode.

Feature Fusion: Feature aggregation: In addition to the Pyramid Pooling Module, PSPNet also includes a join process to efficiently combine multiple features. After pooling, the features are upsampled and combined. This fusion step allows the network to use local and global information. By combining features of different scales, PSPNet improves the network's ability to accurately classify objects of various sizes and shapes. The consolidated feature captures high-quality content and high-level contextual information, improving overall segmentation performance.

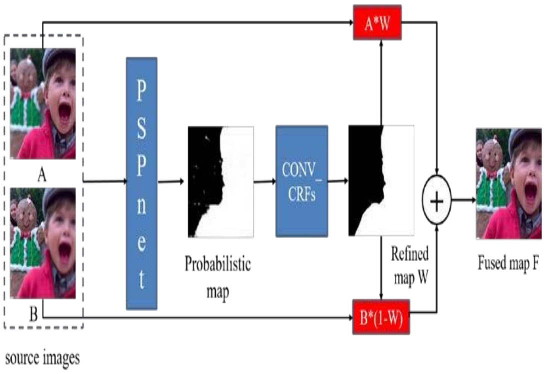


Fig 6.2: Framework of the proposed method for multi-focus image fusion

As shown in Figure 6.2, the plan can be summarized as follows: a multifocal image segmentation method using PSPnet for image compositing is proposed. Unlike the original PSPnet, the auxiliary loss and BCE loss are removed from the network and used. Change in softmax loss to show the final classifier of the pre-trained CNN model Resnet101. It can also be configured as a 6-channel input layer for the first layer of the network, allowing several images to enter the network at the same time; then ConvCRFs are adapted for further processing. Therefore, we consider two multifocal

images of the same location and the fusion process is written as

i. A pair of images A and B is entered into the PSPnet to generate the resulting map of image A, the details of the process are described in Chapter 2.

1 A pair of source images A and B are input to the PSPnet, which is utilized to extract the probabilistic map of image A,and the process details can be seen in section

2. Set ConvCRFs as optimization method to obtain optimization map W of probability map A, see fig. Section 2.2;

3. The final fusion plot F is obtained by the following process:

F A * W B * (1 - W)

Encoder-Decoder Architecture: PSPNet follows an encoder-decoder architecture, which is a common design pattern in semantic segmentation models. The encoder part, often based on a pre-trained convolutional neural network (CNN), extracts hierarchical features from the input image. In PSPNet, the encoder is responsible for capturing low-level visual features. The decoder part, including the Pyramid Pooling Module and feature fusion, takes the extracted features and performs upsampling and fusion operations to generate the final segmentation map. The decoder gradually recovers the spatial resolution and refines the segmented regions. By combining the multi-scale features and leveraging the global and local contextual information, PSPNet achieves accurate semantic segmentation in real-world underwater scenes.

Most semantic segmentation models contains two parts, i.e an Encoder and a Decoder. The Encoder is responsible for the extracting out features from the image, the decoder is the one which predicts the class of the pixel at the end. A typical Encoder-Decoder for segmentation task looks like the architecture shown below:

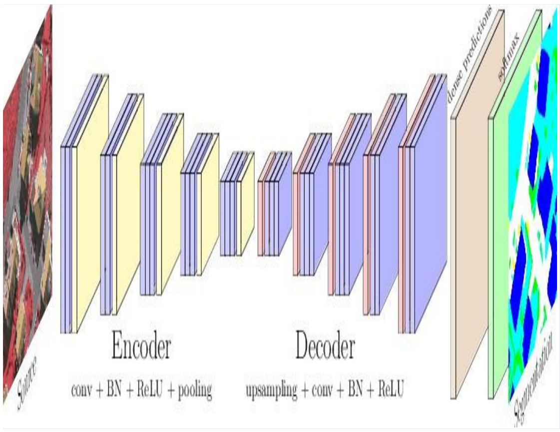


Fig 6.3: Encoder-Decoder Networks for Semantic Segmentation

PSPNet Encoder

The PSPNet encoder contains the CNN backbone with dilated convolutions along with the pyramid pooling module.

Dilated Convolutions

In the last layers of the backbone, we replace the traditional convolutional layers with Dilated convolution layers, which helps in increasing the receptive field. This Dilated convolution layers are placed in the last two blocks of the backbone. Hence the feature recieved at the end of the backbone contains richer features. The illustration[2] shows what dilated convolutions do and how is it different from convolutions.

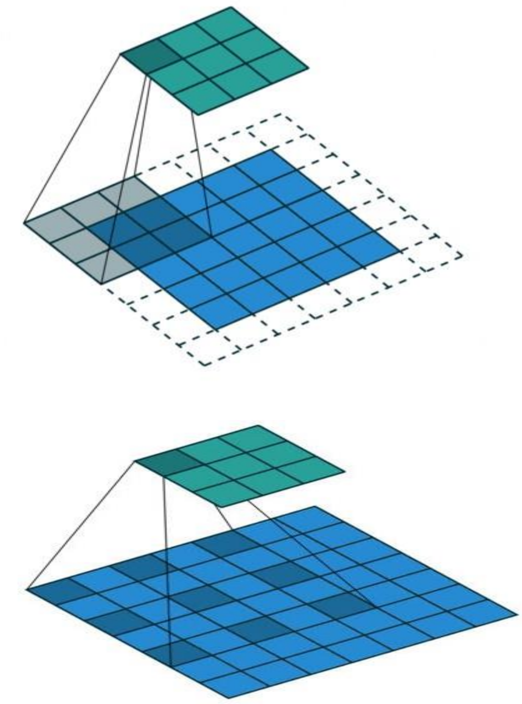


Fig 6.4: Animation of convolution with dilation=2 (left) and dilation=1(right).

In the above fig 4.4, When dilation=1, it is just the standard convolution operation.The value of dilation specifies the sparsity while doing the convolution. We can see that the receptive field for dilated convolution is larger as compared to the standard convolution. The size of the receptive field indicates how much context information we use. In PSPNet, the last two blocks of the backbone have dilation values 2 and 4 respectively.

PSPNet Decoder

After the encoder has extracted out features of the image, it is the turn of the decoder to take those features and convert them into predictions by passing them into its layers. The decoder is just another network which takes in features and results into predictions.

8x upsampling decoder

The PSPNet model is not a complete segmentation model in itself, it is just an encoder, which means it is just half of what is required for image segmentation. The most common decoders that are found in various implementations of PSPNet is a convolution layer followed by a 8x bilinear-upsampling.

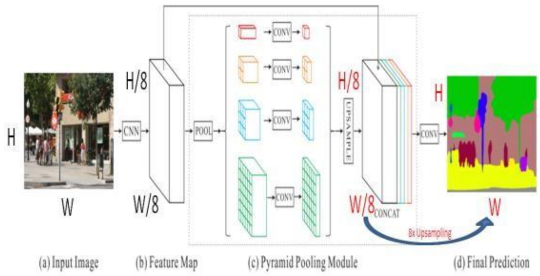


Fig 6.5: PSPNet with 8x upsampling decoder

There is a downside of having a 8x upsmapling decoder in the end is that there are no learnable parameters in them hence the results that we get are blobby and it fails to capture high resolution information from the image.

6.2 : ADVANTAGES AND DISADVANTAGES OF PSPNET

Advantages:

• Multiple data points: PSPNet leverages a wide variety of data points from the pyramid pooling module, allowing the model to understand objects in the environment and improve segmentation accuracy.

• Decision Making: Using pyramid pooling and feature aggregation, PSPNet can generate detailed segmentations with good boundaries by performing accurate product analysis and evaluation.

• Optimization for scaled objects: PSPNet can handle objects of different scales and sizes, adapting to situations where objects may appear from different camera angles or present a big change.

• Strong Semantic Segmentation Performance: PSPNet demonstrates the performance of semantic partitioning tasks and achieves good results on a variety of models and data, including underwater image partitioning

Contextual Reasoning and Global Information Integration: Pyramid pooling and feature aggregation mechanisms allow PSPNet to reason about objects in the entire scene context and combine global information to improve partitioning results.

Multi-field adaptability: PSPNet's design and structure make it applicable to many fields beyond underwater photography. It has been successfully applied to many computer vision tasks such as scene analysis, medical image analysis, and remote sensing.

Disadvantages:

High Computing Requirements: PSPNet's complex architecture is computationally demanding, especially due to the work of pyramid pooling and feature merging. Training and decision making can require significant computing resources and limit its effectiveness on limited hardware.
Large memory space: PSPNet's memory space can be large, especially when processing high resolution images or processing large files. This can cause problems for deployment on devices with limited memory.
Do not rely on large datasets: Like many deep learning models, PSPNet relies on large datasets for training. Obtaining and interpreting diverse and comprehensive data can be time consuming and expensive, especially in specialized fields such as underwater photography.
Susceptibility to noise or poor quality data: PSPNet's performance may suffer for poor quality or underwater images. Poor visibility, noise, and artifacts can adversely affect segmentation accuracy by requiring additional steps or data augmentation techniques.
Interpretation and Interpretation: Deep learning models, including PSPNet, are often criticized for their lack of interpretation and explanation. The complexity of the model makes the decision difficult to understand and explain, which may limit its usefulness in some applications where interpretation is important.
Limitation of the ability to blind cases: When applied to blind cases or areas different from the training data, the performance of the PSPNet may degrade.
Adapting to a new environment or managing change may require additional strategies such as adaptive or adaptive learning.

6.3 : CONCLUSION AND FUTURE WORK :

A layered layer fusion semantic segmentation method PSPNet based on the pyramid pool, which can reduce the error of the model, is proposed. After the images are extracted by the backbone feature extraction network, overall fine point information is obtained with the pyramid pool, and the shallow features of the similarity are further fused during the decoding process to support the information of the feature map. In the feature extraction process, together with the channel display mechanism and the spatial attention mechanism, different parts of the feature map are given weight, the expression of the features is reliable, and the perception of the world features is improved. achieve the goal. improving the segmentation effect. Experiments show that the proposed PSPNet model can classify high-quality and useful images of publicly available data. The rise of deep learning has led to the rapid growth of computer vision.

Although deep learning segmentation algorithms have solved many segmentation problems, there are still some flaws. In the future, the accuracy and speed of the current model can be further improved through refinement and improvement. In addition, less research has been done on the incorporation of increasingly sophisticated elements in this area. In this case the correct segmentation will be affected and there may be differences in brightness, overlapping of multiple targets etc. will lead to negative consequences. Therefore, the generalization ability of the model should be studied further.

7. Effective Underwater mine detection Techniques using Resnet-50 algorithm

ResNet is a conventional neural network for numerous computer vision applications/tasks that is short for Residual Networks. The major advance with ResNet was that we were able to effectively train incredibly deep neural networks with over 150 layers.
Extensive training of neural networks was extremely hard because of the disappearance problem before ResNet. The notion of skip connection was originally developed. The following graphic shows the connection skipping. The picture on the right, we always build convolution layers like before, but now we also add the original input to the output.The picture on the left stacks the layers of convolution one by one. This is called skip connection.

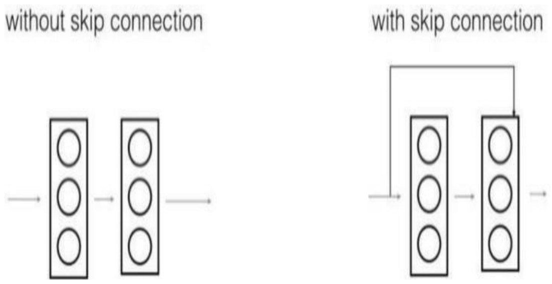


Fig 7.1: Skip Connections in resnet 50

Two lines of code can be written. X shape = X # X Store in a variable the initial value of X. ## Convert + batch standards on X. ## X = Add() # SKIP Connection ([X, X shortcut]). The coding is straightforward but one crucial aspect is that, given that X, X shortcut above aretwo matrices, we can only add them if they have the same form. Then, if the operations of the Convolution + batch standard are performed to the same output form, we may add them simply.

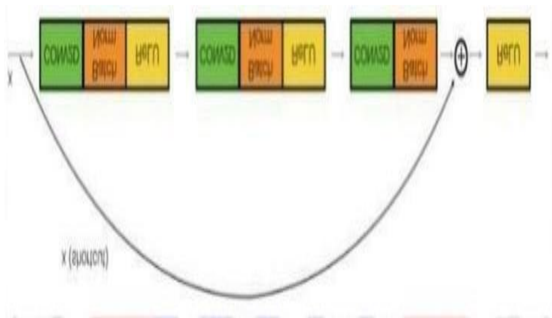


Fig 7.2: Resnet Architecture explaining how skip connections work.

7.1 : LITERATURE SURVEY

The detection of mines has made extensive use of underwater image processing. Utilising autonomous underwater vehicles (AUVs), this is accomplished. These AUVs are placed in the mine detection area to collect mine location data. The base receives the obtained data and takes the appropriate action there.Underwater camera sensors are used by these autonomous vehicles. These sensors have significant disadvantages. Due to noise and scattering, the AUV's underwater images are not accurate. The light transit properties of water and the biological activity on the sea floor are the main causes of this phenomenon.

It is annoying to use underwater optical imaging. When compared to regular photography, there are a lot of challenges. The scattering effect blurs underwater images. Due to wavelength absorption, the colour is diminished. The image that was obtained has noise and water traces. Since artificial lighting is not used in underwater photography, the quality of the image is diminished. In the daytime, the flicker cannot be avoided.

Side-Scan Sonar was the subject of research. The network is supplied the sonar signatures right here. For surveying, this method maps a substantial portion of the ocean floor. The data are first processed by the signal before the training method is applied to the network. After training, the sonar image is separated into subframes during segmentation. Then, using feature extraction, the object property is determined, and the data set is divided.

Mine detection is a barrier for side-scan sonar imaging. The environment in which the system is deployed may change. The image's accuracy suffers as a result. The target mines have a variety of shapes, which the side scan sonar finds challenging to handle. It is possible to misidentify underwater habitations like coral and reefs. These can even conceal things that resemble mines. The sonar signal's slow detection time makes it also unreliable. The side scan sonar maps the target object after sending a sonar wave to it in order to collect data. The process takes a long time, and side scan sonar is very imprecise in rough ocean conditions.

The concept of transformable template matching can also be used to find underwater objects. With this technique, feature extraction is done by creating a templet out of sonar video clips. This is accomplished by analysing acoustic shadows and identifying areas. The target region is found using a method called fast saliency detection. The following step is to extract the normalised gradient feature, which is followed by calculating how similar the target and template are.

Studying the Threshold-based approach revealed that it has historically been widely employed. For mine detecting purposes, this model is not the most effective. The active contour model in the model requires a lot of calculation, and it has an impact on the initial contour.

The sonar image of the model is subpar. Accurate recognition is not enabled by noise and deceptive targets. Towards the Future The integral-image representation is used in sonar imagery. In a short period of time, this offers competitive features. There is a significant reduction in computational load. Small areas of the image are the focus of the piece. The real-time object detecting capabilities of this technique are good. The increased demand for real-time signal processing makes it difficult to detect. There is no density filtering in the method. Here, the algorithm must disregard the fact that multiple mines will be located close to one another.

Mine-cast shadows that resemble other objects are also picked up.

Another technique for underwater image detection is multi beam sonar image processing. This technique makes advantage of the BluView (BV) Sonar. Real-time data is gathered, turned into a picture, and then preprocessed. The foreground and background are distinguished via the contour detection technique. The algorithm used in this process tracks objects. The particle filter tracking approach is used to achieve this. The tracking strategy uses an adaptive fusion tracking strategy. The limited contrast and considerable noise in sonar images make the method problematic.

Another intriguing technique for researching the underwater objects is the adaptive fuzzy neural network. Here, pattern recognition and feed forward are employed. A key tool utilised in its creation is MATLAB. It computes texture features. Features like autocorrelation, total variance, sum average, and sum entropy are used to categorise objects. The model is tested for classification after the texture feature has been trained.

Sensors with monocular vision are used to find objects beneath water. In this technology, light transmission is utilised.The global contrast feature is used to identify the Region of Interest (ROI). To construct the dataset, a monocular camera was employed. The testing model is advanced as various photos are produced. It can reduce noise and improve the system's precision. The approach has shortcomings. The employed camera experiences intensity degradation. The colour distortion and hazy effect compound this drawback.

7.2 Effective Underwater mine detection Techniques using VGG16 algorithm

VGG16 is the architecture of the CNN. VGG16 Until now, it has been considered one of the best vision model designs. Most unusual about VGG16 is that they concentrated on having 3x3 filter coalescing layers with step 1, and always have the same padding and maxpool layer of 2x2 step filter 2 instead of having a lot of hyper parameters. It is included throughout the design of this combination of convolution and max pool layers. It finally has two FC layers (completely linked) and a softmax for output. The 16 in VGG16 refersto a weight of 16 layers. The network is a big network that has around 138 million (about) parameters.

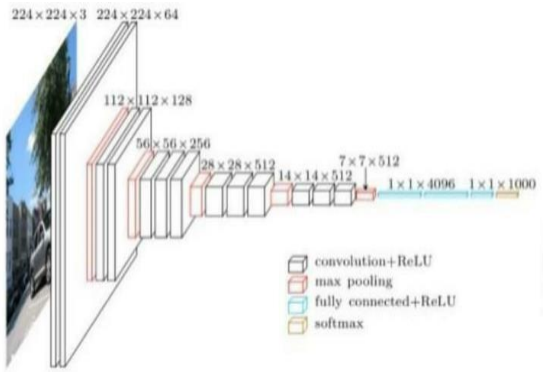


Fig 7.3: Architecture of VGG16

The layer input to a conv is 224 × 224 RGB picture. This picture passes through a stack of convolutionary (conv.) layers, in which a tiny reception field is employed with filters:3×3 (which is the smallest size to capture the notion of left/right, up/down, center). It also uses 1 di 1 convolution filters in one configuration that may be considered as a linear change of the input channels (followed by non-linearity). The tapered strand is set at 1 pixel; the spatial padding is done using the input of the core layer such that after tapering, the spatial resolutionis kept. Five levels of max pooling which follow some of the levels are spatial pooling (notall the conv.layers are followed by max-pooling). A 2-2 pixel window with step 2 allows max pooling.Three fully connected (FC) layers follow a stack of convolutive (different) layers . The lastlayer is the layer of soft-max. The completely linked layers in all networks are the same configuration. The non-linearity of all buried layers (ReLU) is provided. It is also highlighted that none (save one) of the networks have Local Answer Normalization (LRN), nor does such standardization improve ILSVRC data set performance, but leads to increased memory consumption and calculation time.

Table 7.1: Summary of MLO detection and classification methods reviewed in this article.

Application	Authors	Year	Technique	Remarks
Detection Classification	Tucker et al.	2007	Classical image processing	Employments canonical relationship investigation Tall exactness (88%).
Detection	Barngrover et al.	2016	ML	Employments brain–computer interface, Haar-like highlight classifier and SVM.
Detection	Thanh et al.	2020	DL	Uses Gabor-based detector Achieves competitive performance compared to the existing approaches.
Detection	Attaf et al.	2016	Classical image processing	Livelihoods the adequacy overpowering component analysis Exploits the saliency of sonar pictures.
Detection Classification	Saisan et al.	2008	ML	Treats mine discovery as a two-dimensional question acknowledgment and localisation issue Appears great comes about on benchmarking information made from the mine dataset.
Detection	Abu et al.	2019	ML	Employments a back vector machine over the factual features.Does not require information almost the target's shape or estimate.
Detection	Rao et al.	2009	ML	Utilized for real-time application Great comes about on the database given by the Maritime Surface Warfare Center.
Detection Classification	Reed et al.	2003	Classical image processing	Employments unsupervised Markov irregular field model. Utilises a priori data almost the spatial relationship between highlights and shadows.
Classification	Fei et al.	2015	DL	Uses an ensemble learning scheme in the Dempster–Shafer theory framework
Detection	Acosta et al.	2015	Classical image processing	Employments Cell Average–Constant False Alarm Rate Suitable for independent submerged vehicles

Table 7.1: *Cont.*

Application	Authors	Year	Technique	Remarks
Classification	Neumann et al.	2008	Classical image processing	Employments the Hough change Altogether diminishes the number of wrong discoveries
Classification	McKay et al.	2017	DL	Employments convolutional neural systems and exchange learning
Classification	Yao et al.	2002	DL	Employments include determination and neural arrange classifier Introduces a sub-band combination component for wideband information
Classification	Dobeck et al.	1997	DL	Employments the k-nearest neighbour attractor-based neural network classifier
Detection Classification	Ciany et al.	2003	Classical image processing	Employments signal-to-noise proportion and shape of the objects Suitable for independent submerged vehicles
Detection Classification	Williams et al.	2016	DL	Employments profound systems learned for a few parallel classification errands
Classification	Galusha et al	2019	DL	Employments convolutional neural systems for area based on cross-validation
Detection Classification	Dura et al.	2008	Classical image processing	Employments a superellipse fitting approach. The classification rate is higher than 80%.
Detection	Wu et al.	2019	DL	Employments the productive convolutional arrange engineering for semantic division.

Table 7.1: *Cont.*

Application	Authors	Year	Technique	Remarks
Semantic segmentation	Yanming Guo Yu Liu ·	2017	CNN	Deep learning advancements revolutionize image segmentation, impacting various visual tasks.
Semantic segmentation	Rishipal Singha , Rajneesh Rani	2020	DL AND DCNN	This paper explores DCNNs for semantic segmentation and emphasizes the need for improved real-time performance.
Semantic segmentation	Xiaolong Liu1 · Zhidong Deng	2018	CNN AND FCN	Comprehensive review of recent progress in DCNN-based semantic image segmentation methods.
instance segmentation	Wenchao Gu a , Shuang Bai	20212	DCNN	This paper discusses instance segmentation, including evaluation, methods, backbones, and future directions
instance segmentation	Kuo-Kun Tseng a , Jiangrui Lin	20212	CNN	A fast instance segmentation algorithm combining YOLOv3 and Mask-RCNN is proposed, with improved speed and accuracy.
instance segmentation	Farhana Sultana a , Abu Sufia	20202	DCNN AND CNN	This article provides an overview of image segmentation models, their optimization, and application areas
instance segmentation	Hao Chena,∗ , Xiaojuan Q	2016	Object detection	Deep contour-aware network improves histological object segmentation with superior performance.

8. CONCLUSION AND FUTURE WORK

In summary, U-Net, Fully Convolutional Network (FCN), Semantic Segmentation, PSPNet (Pyramid Scene Parsing Network) and Image Segmentation are all important and useful concepts in the field of computer vision, especially later in image segmentation.

U-Net is a powerful tool for image segmentation, especially in the field of biomedical imaging, as it can capture local and global features by shrinking and expanding. It excels in tasks that require a clear definition of the structure and can be adapted to many different situations with the necessary modifications.
FCN, including

U-Net, revolutionized image segmentation by providing end-to-end pixel-level prediction. They replace all layers with convolutional layers, so spatial information is preserved.
FCNs have proven effective in many applications and are widely used as the base model for semantic segmentation tasks.

Semantic segmentation aims to assign a semantic tag to each pixel in an image and provide pixel-level detailed information about the image. FCNs with U-Net are often used in semantic segmentation tasks because of their ability to capture spatial information and learn discrimination.

PSPNet (Pyramid Scene Parsing Network) is another well-known semantic segmentation architecture. It includes a pyramid pooling module to capture multiple data points, providing more powerful and accurate segmentation.
PSPNet performs competitively on a variety of metrics and is particularly useful for capturing global content.

Determining which of these methods is better depends on specific tasks, data characteristics, and performance measures. U-Net is advantageous where accurate localization and detailed boundary information are important. FCNs, including U-Net, are versatile and can be widely used in many fields. PSPNet is good at capturing the global context, which is important for projects that require a comprehensive understanding of the situation.
Ultimately, the choice of the most appropriate method depends on the specific requirements of the application and the balance between accuracy, efficiency and budget. It is recommended to evaluate and compare these project datasets and project models to determine the best approach for a given situation.

Regarding future work of U-Net, FCN, Semantic Segmentation, PSPNet, and Image Segmentation, there are several areas that researchers can focus on to further develop these models:

Architectural Enhancements: One of the aspects of future work is to explore Architecture enhancements and enhancements. This may include combining monitoring systems, spatial and channel monitoring modules, or cross-linking to improve data flow and performance.

Efficiency and Speed: Although this system performs well, there are still areas for improvement in terms of efficiency and speed.
Future work may focus on developing more robust modeling techniques or exploring techniques such as network convolution and quantization to reduce model size and speed up inference time.

Domain Adaptation and Generalization: Developing the general capabilities of these processes is important for practical applications. Future studies may explore transfer and transfer learning strategies to improve the performance of the model on different data and different image formats.

Weakly Supervised and Semi-supervised Learning: Training deep learning models for image segmentation often requires a lot of recording data. Future work may explore unsupervised and semi-supervised learning strategies where models can use limited or incomplete explanations to better perform partitioning tasks.
Understanding context: While semantic segmentation focuses on pixel-level classification,

combined with higher context understanding can improve performance. Future research may explore ways to capture long-term dependencies and integrate data points from different scales and abstraction levels.

Deciding which method is better between U-Net, FCNs, semantic segmentation and PSPNet depends on the specific task, dataset and evaluation criteria. Each method has advantages and limitations. It is recommended to evaluate and compare these project data and project models to determine which method is more accurate, efficient and suitable for the specific applications designed.
Overall, future work in these areas aims to advance image segmentation by solving current limitations and challenges, well improving model performance, and extending the applicability of this process to many places and situations around the world.

9.REFERENCES

1. Del Rio Vera, J.; Coiras, E.; Groen, J.; Evans, B. Automatic Target Recognition in Synthetic Aperture Sonar Images Based on Geometrical Feature Extraction. *EURASIP J. Adv. Signal Process.* 2009, *2009*, 109438. https://doi.org/10.1155/2009/109438.

2. Dura, E.; Bell, J.; Lane, D. Superellipse Fitting for the Recovery and Classification of Mine-Like Shapes in Sidescan Sonar Images. *IEEE J. Ocean. Eng.* 2008, *33*, 434–444. https://doi.org/10.1109/JOE.2008.2002962.

3. Neupane, D.; Seok, J. A review on deep learning-based approaches for automatic sonar target recognition. *Electronics* 2020, *9*, 1972.

4. Barngrover, C.; Kastner, R.; Belongie, S. Semisynthetic versus real-world sonar training data for the classification of mine-like objects. *IEEE J. Ocean. Eng.* 2015, *40*, 48–56. https://doi.org/10.1109/JOE.2013.2291634.

5. Cerqueira, R.; Trocoli, T.; Neves, G.; Joyeux, S.; Albiez, J.; Oliveira, L. A novel GPU-based sonar simulator for real-time applications. *Comput. Graph.* 2017, *68*, 66–76. https://doi.org/10.1016/j.cag.2017.08.008.

6. Borawski, M.; Forczmański, P. Sonar Image Simulation by Means of Ray Tracing and Image Processing. In *Enhanced Methods in Computer Security, Biometric and Artificial Intelligence Systems*; Kluwer Academic Publishers: Boston, MA, USA, 2005; pp. 209–214.

7. Saeidi, C.; Hodjatkashani, F.; Fard, A. New tube-based shooting and bouncing ray tracing method. In Proceedings of the 2009 International Conference on Advanced Technologies for Communications, Hai Phong, Vietnam, 12–14 October 2009; pp. 269–273.

8. Danesh, S.A. *Real Time Active Sonar Simulation in a Deep Ocean Environment*; Massachusetts Institute of Technology: Cambridge, MA, USA, 2013.

9. Saito, H.; Naoi, J.; Kikuchi, T. Finite Difference Time Domain Analysis for a Sound Field Including a Plate in Water. *Jpn. J. Appl. Phys.* 2004, *43*, 3176–3179. https://doi.org/10.1143/JJAP.43.3176.
10. Maussang, F.; Rombaut, M.; Chanussot, J.; Hétet, A.; Amate, M. Fusion of local statistical parameters for buried underwater mine detection in sonar imaging. *EURASIP J. Adv. Signal Process.* 2008, *2008*, 1–19. https://doi.org/10.1155/2008/876092.
11. Maussang, F.; Chanussot, J.; Rombaut, M.; Amate, M. From Statistical Detection to Decision Fusion: Detection of Underwater Mines in High Resolution SAS Images. In *Advances in Sonar Technology*; IntechOpen: London, UK, 2009; ISBN 9783902613486.
12. Lurton, X. *An Introduction to Underwater Acoustics*; Springer: Berlin/Heidelberg, Germany, 2010; ISBN 978-3-540-78480-7.
13. Barngrover, C.M. *Automated Detection of Mine-Like Objects in Side Scan Sonar Imagery*; University of California: San Diego, CA, USA, 2014.
14. Tellez, O.L.L.; Borghgraef, A.; Mersch, E. The Special Case of Sea Mines. In *Mine Action—The Research Experience of the Royal Military Academy of Belgium*; InTechOpen: London, UK, 2017; pp. 267–322.
15. Doerry, A. Introduction to Synthetic Aperture Sonar. In Proceedings of the 2019 IEEE Radar Conference (RadarConf), Boston, MA, USA, 22–26 April 2019; pp. 1–90.
16. Atherton, M. *Echoes and Images, The Encyclopedia of Side-Scan and Scanning Sonar Operations*; OysterInk Publications: Vancouver, BC, Canada, 2011; ISBN 098690340X.
17. Rao, C.; Mukherjee, K.; Gupta, S.; Ray, A.; Phoha, S. Underwater mine detection using symbolic pattern analysis of sidescan sonar images. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 5416–5421. https://doi.org/10.1109/ACC.2009.5160102.
18. Tucker, J.D.; Azimi-Sadjadi, M.R.; Dobeck, G.J. Canonical Coordinates for Detection and Classification of Underwater Objects From Sonar Imagery. In Proceedings of the OCEANS 2007—Europe, Aberdeen, UK, 18–21 June 2007; pp. 1–6.
19. Reed, S.; Petillot, Y.; Bell, J. An automatic approach to the detection and extraction of mine features in sidescan sonar. *IEEE J. Ocean. Eng.* 2003, *28*, 90–105. https://doi.org/10.1109/JOE.2002.808199.
20. Klausner, N.; Azimi-Sadjadi, M.R.; Tucker, J.D. Underwater target detection from multi-platform sonar imagery using multichannel coherence analysis. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 2728–2733.
21. Langner, F.; Knauer, C.; Jans, W.; Ebert, A. Side scan sonar image resolution and automatic object detection, classification and identification. In Proceedings of the OCEANS '09 IEEE Bremen: Balancing Technology with Future Needs, Bremen, Germany, 11–14 May 2009; pp. 1–8.
22. Hożyń, S.; Zalewski, J. Shoreline Detection and Land Segmentation for Autonomous Surface Vehicle Navigation with the Use of an Optical System. *Sensors* 2020, *20*, 2799. https://doi.org/10.3390/s20102799.
23. Hożyń, S.; Żak, B. Local image features matching for real-time seabed tracking applications. *J. Mar. Eng. Technol.* 2017, *16*, 273–282. https://doi.org/10.1080/20464177.2017.1386266.
24. Wang, X.; Wang, H.; Ye, X.; Zhao, L.; Wang, K. A novel segmentation algorithm for side-scan sonar imagery with multi-object. In Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, 15–18 December 2007; pp. 2110–2114.
25. Om, H.; Biswas, M. An Improved Image Denoising Method Based on Wavelet Thresholding. *J. Signal Inf. Process.* 2012,
26.Hoz ̇yn ́,S.;Z ̇ak,B.SegmentationAlgorithmUsing MethodofEdgeDetection.SolidStatePhenom.2013,196,206–211.[CrossRef]
27. Celik, T.; Tjahjadi, T. A Novel Method for Sidescan Sonar Image Segmentation. IEEE J. Ocean. Eng. 2011, 36, 186–194. [CrossRef]
28. Wei, S.; Leung, H.; Myers, V. An automated change detection approach for mine recognition using sidescan sonar data. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 553–558.
29. Neumann, M.; Knauer, C.; Nolte, B.; Brecht, D.; Jans, W.; Ebert, A. Target detection of man made objects in side scan sonar images segmentation based false alarm reduction. J. Acoust. Soc. Am. 2008, 123, 3949. [CrossRef]
30. Huo, G.; Yang, S.X.; Li, Q.; Zhou, Y. A Robust and Fast Method for Sidescan Sonar Image Segmentation Using Nonlocal Despeckling and Active Contour Model. IEEE Trans. Cybern. 2017, 47, 855–872. [CrossRef]

31. Acosta, G.G.; Villar, S.A. Accumulated CA–CFAR Process in 2-D for Online Object Detection From Sidescan Sonar Data. IEEE J. Ocean. Eng. 2015, 40, 558–569. [CrossRef]
32. Ye, X.-F.; Zhang, Z.-H.; Liu, P.X.; Guan, H.-L. Sonar image segmentation based on GMRF and level-set models. Ocean Eng. 2010, 37, 891–901. [CrossRef]
33. Fei, T.; Kraus, D. An expectation-maximisation approach assisted by Dempster-Shafer theory and its application to sonar image segmentation. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 1161–1164. [CrossRef]
34. Szymak, P.; Piskur, P.; Naus, K. The Effectiveness of Using a Pretrained Deep Learning Neural Networks for Object Classification in Underwater Video. Remote Sens. 2020, 12, 3020. [CrossRef]
35. Huo, G.; Wu, Z.; Li, J. Underwater Object Classification in Sidescan Sonar Images Using Deep Transfer Learning and Semisynthetic Training Data. IEEE Access 2020, 8, 47407–47418. [CrossRef]
36. Attaf, Y.; Boudraa, A.O.; Ray, C. Amplitude-based dominant component analysis for underwater mines extraction in side scans sonar. In Proceedings of the Oceans 2016—Shanghai, Shanghai, China, 10–13 April 2016. [CrossRef]
37. Wu, M.; Wang, Q.; Rigall, E.; Li, K.; Zhu, W.; He, B.; Yan, T. ECNet: Efficient Convolutional Networks for Side Scan Sonar Image Segmentation. Sensors 2019, 19, 2009. [CrossRef] [PubMed]
38. Abu, A.; Diamant, R. A Statistically-Based Method for the Detection of Underwater Objects in Sonar Imagery. IEEE Sens. J. 2019, 19, 6858–6871. [CrossRef]
39. McKay, J.; Monga, V.; Raj, R.G. Robust Sonar ATR Through Bayesian Pose-Corrected Sparse Classification. IEEE Trans. Geosci. Remote Sens. 2017, 55, 5563–5576. [CrossRef]
40. Williams, D.P. Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2497–2502. [CrossRef]
41. Ciany, C.M.; Zurawski, W.C.; Dobeck, G.J.; Weilert, D.R. Real-time performance of fusion algorithms for computer aided detection and classification of bottom mines in the littoral environment. In Proceedings of the Oceans 2003. Celebrating the Past . . . Teaming Toward the Future (IEEE Cat. No.03CH37492), San Diego, CA, USA, 22–26 September 2003; Volume 2, pp. 1119–1125.
42. Saisan, P.; Kadambe, S. Shape normalised subspace analysis for underwater mine detection. In Proceedings of the 2008 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 1892–1895.
43. Thanh Le, H.; Phung, S.L.; Chapple, P.B.; Bouzerdoum, A.; Ritz, C.H.; Tran, L.C. Deep gabor neural network for automatic detection of mine-like objects in sonar imagery. IEEE Access 2020, 8, 94126–94139. [CrossRef]

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

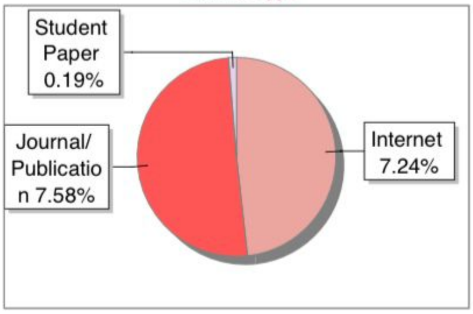
Submission Information

Author Name	moin
Title	paper4
Paper/Submission ID	757393
Submission Date	2023-05-31 08:37:44
Total Pages	18
Document type	Research Paper

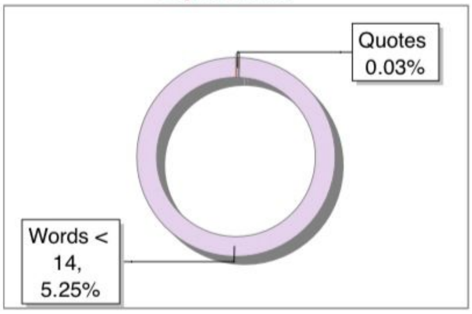
Result Information

Similarity **15 %**

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

SIMILARITY %	MATCHED SOURCES	GRADE	
15	63	B	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.mdpi.com	2	Internet Data
2	towardsdatascience.com	2	Internet Data
3	Emerging artificial intelligence methods in structural engineering by Salehi-2018	1	Publication
4	A deep neural networks based model for uninterrupted marine environment monitori by G-2020	1	Publication
5	Encoder-decoder CNN models for automatic tracking of tongue contours in real-tim by Hame-2020	1	Publication
6	A Comparison Different DCNN Models for Intelligent Object Detection i by Ding-2018	<1	Publication
7	A review of feature selection techniques in sentiment analysis, by Ahmad, Siti Rohaida- 2019	<1	Publication
8	Hierarchical Features Driven Residual Learning for Depth Map Super-Re, by Guo, Chunle Li, Ch- 2018	<1	Publication
9	IEEE 2011 IEEE International Conference on Computer Vision Workshops	<1	Publication
10	Explosives Detection by Terahertz SpectroscopyA Bridge Too Far by Kemp-2011	<1	Publication
11	www.acm.org	<1	Internet Data

12	www.atlantis-press.com	<1	Publication
13	www.projectpro.io	<1	Internet Data
14	hess.copernicus.org	<1	Publication
15	docplayer.net	<1	Internet Data
16	IEEE 2018 25th National and 3rd International Iranian Conference on, by Mohammadi, Elnaz O- 2018	<1	Publication
17	arxiv.org	<1	Publication
18	Pattern Recognition and Classification Introduction, by Dougherty, Geoff- 2013	<1	Publication
19	malaevents.in	<1	Internet Data
20	surge.iitk.ac.in	<1	Publication
21	dochero.tips	<1	Internet Data
22	moam.info	<1	Internet Data
23	bmcgenomics.biomedcentral.com	<1	Internet Data
24	deepai.org	<1	Internet Data
25	arxiv.org	<1	Publication
26	www.ncbi.nlm.nih.gov	<1	Internet Data
27	IEEE 2017 9th International Conference on Information Technology an, by Fazekas, Szilard Zs- 2017	<1	Publication
28	IEEE 2017 IEEE 25th Annual Symposium on High-Performance Interconne, by Lu, Xiaoyi Shi, Ha- 2017	<1	Publication
29	1library.co	<1	Internet Data

30	Advances in Machine Learning and Data Science by Damoda-2018	<1	Publication
31	DisepNet for breast abnormality recognition by Yu-2021	<1	Publication
32	ijrte.org	<1	Publication
33	koreascience.or.kr	<1	Publication
34	towardsdatascience.com	<1	Internet Data
35	www.doaj.org	<1	Publication
36	www.researchgate.net	<1	Internet Data
37	www.sciencepubco.com	<1	Publication
38	Characterization of lenticulostriate arteries with high resolution black-blood T by Ma-2019	<1	Publication
39	mdpi.com	<1	Internet Data
40	mdpi.com	<1	Internet Data
41	qdoc.tips	<1	Internet Data
42	Solving the over segmentation problem in applications of Watershed Transform by Gonzalez-2013	<1	Publication
43	www.dx.doi.org	<1	Publication
44	IEEE 2018 10th International Conference on Knowledge and Systems En, by Nguyen, Mau Uyen D- 2018	<1	Publication
45	IEEE 2018 Eighth International Conference on Image Processing Theor, by Jiang, Xiaoyue Du,- 2018	<1	Publication
46	Deep learning based automatic detection of uninformative images in pulmonary opt by Brochet-2020	<1	Publication

47	mdpi.com	<1	Internet Data
48	citeseerx.ist.psu.edu	<1	Internet Data
49	www.indiamart.com	<1	Internet Data
50	afflictor.com	<1	Internet Data
51	arxiv.org	<1	Publication
52	Demand for Fresh Vegetables in the United States 19702010	<1	Student Paper
53	docplayer.net	<1	Internet Data
54	Effective Deep Learning Approaches for Summarization of Legal Texts by Anand-2019	<1	Publication
55	ijcea.com	<1	Publication
56	Lifetime stress exposure and health A review of contemporary assessment methods by Shields-2017	<1	Publication
57	moam.info	<1	Internet Data
58	Semi-supervised learning with generative model for sentiment classification of s by Duan-2020	<1	Publication
59	The Future of Quality Measurement in the United States by Yi-2014	<1	Publication
60	towardsdatascience.com	<1	Internet Data
61	Wavelets for texture analysis an overview	<1	Student Paper
62	www.atlantis-press.com	<1	Internet Data
63	www.mecs-press.org	<1	Publication