International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Real Time Recognition of Underwater Images Using Deep Learning

Dhiresh Kharga[1], Ruchira C[2]

[1,2]8th Semester, BE, BNMIT

Abstract

In order to tackle the challenges of the underwater environment, data augmentation techniques are employed during training to increase model diversity and robustness. This involves augmenting the dataset with variations in lighting, blur, and distortion, enabling the model to generalize effectively to unseen underwater scenarios. The image recognition pipeline consists of preprocessing, feature extraction, and classification stages. Preprocessing techniques enhance image quality, reduce noise, and correct color distortion caused by water absorption. Feature extraction is performed using specially designed Support Vector Classifier (SVC) architectures for underwater imagery, allowing the network to learn meaningful representations. The trained model is evaluated on a separate test set, demonstrating its effectiveness in detecting and localizing humans in challenging underwater conditions. This approach has promising applications in underwater surveillance, search and rescue operations, and marine biology research, facilitating automated analysis and decision- making processes for advancements in underwater exploration and monitoring technologies.

CHAPTER 1 INTRODUCTION

CHAPTER 1

INTRODUCTION

Deep learning represents a revolutionary approach in the realm of artificial intelligence, mirroring the intricate workings of the human brain to process data. Through complex algorithms and neural network architectures, deep learning models possess the remarkable ability to discern intricate patterns within various forms of data, including images, text, audio, and more. This capability enables them to extract meaningful insights and make accurate predictions, often surpassing traditional machine learning methods in tasks requiring nuancedunderstanding and abstraction.

When applied to underwater image processing, deep learning emerges as a powerful tool for enhancing the quality and interpretability of images captured beneath the water's surface. This specialized field leverages deep neural networks to tackle a range of tasks crucial for underwater exploration and analysis. Among these tasks, denoising stands out as a fundamental process aimed at reducing or eliminating unwanted noise from images, thereby improving clarity and fidelity. In the murky depths where visual information is often obscured by environmental factors such as water turbidity and particulate matter, denoising algorithms play a vital role in revealing the true details of underwater scenes.

Moreover, deep learning facilitates color correction techniques tailored to the unique challenges of underwater imaging. The absorption and scattering of light in water can distort color representations, leading to inaccurate or muddied visuals. Through sophisticated learning algorithms, deep neural networks can automatically adjust color balances, restoring natural hues and enhancing the overall perceptual quality of underwater images.

Furthermore, object recognition represents a crucial aspect of underwater image processing, enabling automated identification and classification of various marine organisms, structures, or artifacts. Deep learning models excel in this task by learning intricate feature representations and discerning subtle visual cues, even amidst challenging underwater environments characterized by fluctuating lighting conditions and occlusions.

In essence, the fusion of deep learning techniques with underwater image processing heralds a new era of exploration and understanding in marine science and engineering. By harnessing the computational prowess of neural networks, researchers and practitioners can unlock invaluable insights from the depths of the ocean, unveiling hidden mysteries and paving the way for innovative applications in fields ranging from marine biology to underwater archaeology and beyond.

1.1 MOTIVATION

The motivation for undertaking a project focused on underwater image processing using deep learning stems from several compelling factors:

- Scientific Exploration: The oceans cover more than 70% of the Earth's surface, yet much of the underwater world remains unexplored and mysterious. By enhancing our ability to process and interpret underwater images, we can unlock valuable insights into marine ecosystems, geological formations, and underwater habitats. This knowledge is essential for understanding Earth's biodiversity, monitoring environmental changes, and informing conservation efforts.
- Technological Innovation: Deep learning represents a cutting-edge technology with vast potential for transformative applications. By applying deep neural networks to underwater image processing, we can leverage the latest advancements in artificial intelligence to address longstanding challenges in marine science and engineering. This not only drives innovation within the field but also opens up new avenues for interdisciplinary collaboration and cross-sectoral partnerships.
- Environmental Monitoring and Management: With growing concerns about the health of our oceans and the impacts of climate change, there is an urgent need for effective environmental monitoring and management strategies. By improving the quality and accessibility of underwater imagery, deep learning-enabled image processing systems can support efforts to monitor marine ecosystems, track changes over time, and assessthe effectiveness of conservation measures.
- Resource Exploration and Management: The oceans are a rich source of valuable resources, including fish stocks, minerals, and energy reserves. By deploying autonomous underwater vehicles (AUVs) equipped with deep learning-based image processing capabilities, researchers and industry stakeholders can streamline the exploration and exploitation of marine resources while minimizing environmental impacts and ensuring sustainable management practices.
- Safety and Security: Underwater imaging technologies play a crucial role in various domains, including maritime security, underwater surveillance, and search and rescue operations. By enhancing the clarity and interpretability of underwater images through deep learning techniques, we can improve situational awareness, facilitate rapid decision-making, and enhance the safety and security of maritime activities.
- Overall, the project topic of underwater image processing using deep learning is motivated by a desire to unlock the potential of the underwater world, advance scientific knowledge, foster technological innovation, and address pressing environmental and societal challenges facing our

oceans.

1.2 PROBLEM STATEMENT

Given a set of underwater images with varying degrees of noise, distortion, and lighting conditions, the task is to develop an accurate and efficient image recognition system using artificial neural networks. The system should be able to classify the images into predefined categories such as different species of marine organisms, underwater objects, or environmental conditions.

1.3 OBJECTIVE

In pursuit of advancing our understanding and utilization of underwater environments, the project aims to harness the power of deep learning for enhancing the quality and interpretability of underwater imagery in the following ways :

- Conservation Efforts: Image recognition can be used to identify different species of marine life, which can help conservation efforts. By recognizing different types of marine creatures and their behaviours, researchers can better understand the ocean ecosystem and make more informed decisions about conservation measures.
- Environmental Monitoring: Underwater images can be used to monitor the health of the ocean, including the impact of pollution, climate change, and other environmental factors. Image recognition technology can help automate this process by identifying changes in the underwater environment, such as the presence of harmful algal blooms or changes in water temperature Search and Rescue: Image recognition can be used to locate lost or missing people in the water. By analysing underwater images, rescue teams can identify objects or people that are not visible to the naked eye, which can greatly improve search and rescue efforts.
- Commercial and Recreational Activities: Image recognition can also be used for commercial and recreational activities such as fishing, scuba diving, and ocean tourism. By analysing underwater images, businesses can better understand the behaviour of marine creatures and make more informed decisions about where to fish or how to interact with marine life.

1.4 SUMMARY

Once the model has been refined and optimized, it is ready for deployment in a real-world underwater environment. This stage involves setting up specialized hardware and communication systems to facilitate the model's operation. The deployment may include integrating the model into underwater robotics or surveillance systems, allowing it to analyze and detect human bodies in real-time, contributing to applications such as underwater surveillance, search and rescue operations, or marine biology research.

CHAPTER 2 LITERATURE SURVEY

CHAPTER 2
LITERATURE SURVEY

As we embark on our journey to delve into the existing literature, it's paramount to contextualize the profound significance of prior research within the dynamic realm of Underwater Image Detection. Understanding the historical trajectory and current state of scholarly inquiry not only provides valuable insights into the evolution of ideas but also illuminates the pressing questions and challenges that continue to shape the discourse in this domain.

By conducting a comprehensive survey of the literature, we aim to traverse the intellectual landscape, discerning seminal contributions, emerging trends, and methodological innovations. This process not only facilitates a deeper appreciation of the complexities inherent in our research topic but also enables us to critically evaluate existing knowledge gaps and opportunities for advancement.

Moreover, by situating our own investigation within the broader context of existing scholarship, we endeavor to cultivate a nuanced understanding of the theoretical frameworks, empirical findings, and practical implications that underpin our research endeavor. Ultimately, this iterative dialogue with the existing literature serves as a guiding beacon, illuminating our path forward as we navigate the uncharted waters of discovery and innovation.

2.1 OVERVIEW

The literature survey conducted for this project encompasses a comprehensive exploration of scholarly works spanning the interdisciplinary landscape of Underwater Image Detection. Through meticulous review and analysis, we have synthesized a wealth of existing research, ranging from seminal contributions to recent advancements, in order to contextualize our own investigation and identify key insights, gaps, and methodologies.

Our survey begins by tracing the historical trajectory of inquiry, elucidating foundational concepts and theoretical frameworks that have shaped the discourse within our field. Building upon this foundation, we delve into contemporary scholarship, examining seminal studies, empirical findings, and theoretical debates that underscore the complexities inherent in our research topic.

Furthermore, our review extends beyond traditional disciplinary boundaries, incorporating interdisciplinary perspectives and cross-cutting themes that intersect with our research inquiry. By synthesizing insights from diverse fields, we aim to foster a holistic understanding of the multifaceted dimensions of our research topic and identify innovative approaches for addressing current challenges and advancing knowledge.

Throughout the literature survey, particular attention is paid to identifying recurring themes, theoretical frameworks, and methodological approaches that have garnered consensus or provoked scholarly debate. By critically engaging with existing scholarship, we endeavor to chart a course for our own research that builds upon prior knowledge while pushing the boundaries of inquiry into new and unexplored territories.

Ultimately, the literature survey serves as a foundational pillar of our research endeavor, providing the theoretical scaffolding and intellectual context necessary to inform our research questions, methodologies, and interpretations. By synthesizing insights from existing scholarship, we aspire to contribute novel perspectives, empirical findings, and methodological innovations that advance our understanding and address pressing challenges within our field.

2.2 LITERATURE SURVEY

1. SWIPENET AND CMA (CURRICULUM MULTI-CLASS ADABOOST)

Authors: Long Chen, Feixiang Zhou , Shengke Wang , “SWIPENET: Object detection in noisy underwater scenes”.

Description: WeIghted hyPEr Network (SWIPENET), and a robust training paradigm named Curriculum Multi-Class Adaboost (CMA), to address these two problems at the same time. Firstly, the backbone of SWIPENET produces multiple high resolution and semantic-rich Hyper Feature Maps, which significantly improve small object detection. Secondly, a novel sample-weighted detection loss function is designed for SWIPENET, which focuses on learning high weight samples and ignores learning low weight samples. Moreover, inspired by the human education process that drives the learning from easy to hard concepts, we here propose the CMA training paradigm that first trains a clean detector which is free from the influence of noisy data. Then, based on the clean detector, multiple detectors focusing on learning diverse noisy data are trained and incorporated into a unified deep ensemble of strong noise immunity.

2. FISH DETECTION

Authors: Wenwei Xu, Shari Matzner , "Underwater Fish Detection using Deep Learning for Water Power Applications".

Description: Clean energy from oceans and rivers is becoming a reality with the development of new technologies like tidal and instream turbines that generate electricity from naturally flowing water. These new technologies are being monitored for effects on fish and other wildlife using underwater video. Methods for automated analysis of underwater video are needed to lower the costs of analysis and improve accuracy. A deep learning model, YOLO, was trained to recognize fish in underwater video using three very different datasets recorded at real-world water power sites. Training and testing with examples from all three datasets resulted in a mean average precision (mAP) score of 0.5392. To test how well a model could generalize to new datasets, the model was trained using examples from only two of the datasets and then tested on examples from all three datasets.

The resulting model could not recognize fish in the dataset that was not part of the training set. The mAP scores on the other two datasets that were included in the training set were higher than the scores achieved by the model trained on all three datasets. These results indicate that different methods are needed in order to produce a trained model that can generalize to new data sets such as those encountered in real world applications.

3. ROLMIX

Authors: Wei-Hong Lin, Jia-Xing Zhong, Shan Liu, Thomas Li, Ge Li , " Fusion among Multiple Images for Underwater Object Detection"

Description: Generic object detection algorithms have proven their excellent performance in recent years. However, object detection on underwater datasets is still less explored. In contrast to generic datasets, underwater images usually have color shift and low contrast; sediment would cause blurring in underwater images. In addition, underwater creatures often appear closely to each other on images due to their living habits. To address these issues, our work investigates augmentation policies to simulate overlapping, occluded and blurred objects, and we construct a model capable of achieving better generalization. We propose an augmentation method called RoIMix, which characterizes interactions among images. Proposals extracted from different images are mixed together. Previous data augmentation methods operate on a single image while we apply RoIMix to multiple images to create enhanced samples as training data. Experiments show that our proposed method improves the performance of region-based object detectors on both Pascal VOC and URPC datasets.

4. DEEP LEARNING FOR IMAGE SEGMENTATION: A SURVEY

Authors: Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr

Description: This survey paper delves into the realm of deep learning techniques specifically tailored for image segmentation tasks. Image segmentation involves partitioning an image into multiple segments or regions to simplify its representation and facilitate analysis. The paper comprehensively discusses various deep learning architectures, including Convolutional Neural Networks (CNNs), and their application in image segmentation. It explores different aspects such as semantic segmentation, instance segmentation, and panoptic segmentation, providing insights into their strengths and limitations. Moreover, the paper highlights the significance of image segmentation in numerous fields, including underwater image analysis, where accurate segmentation plays a crucial role in tasks such as object detection and classification.

5. UNDERWATER IMAGE ENHANCEMENT TECHNIQUES: A REVIEW Authors: Mohammed Hassan, Naglaa Fathy, Mennatallah Saad, and Hassan H. Aly

Description: This review paper offers an in-depth examination of techniques aimed at enhancing the quality of underwater images. Underwater imaging poses unique challenges due to factors like attenuation, scattering, and color distortion, which degrade image quality. To address these challenges, various image enhancement techniques have been developed. The paper categorizes these techniques into single-image and multi-image approaches, discussing their principles, methodologies, and performance characteristics. Single-image techniques typically involve color correction, contrast enhancement, and noise reduction, while multi-image techniques leverage multiple images to improve clarity and visibility. The review also discusses the effectiveness of these techniques in mitigating underwater image degradation and their applicability in real-world scenarios, including underwater exploration, surveillance, and marine biology research.

2.3 APPROACH TOWARDS THE PROBLEM

Object detection in computer vision helps identify and locate things in images or videos by creating bounding boxes around them. It's used for counting items, tracking them in real- time, and labeling them accurately. Deep learning-based object detection models typically have two parts: an encoder that extracts features from images, and a decoder that determines bounding boxes and labels. There are different decoder methods like pure regression and region proposal networks. The accuracy of object detection is often measured using intersection-over-union (IOU), which compares the predicted bounding boxes with the actual ones.

SVC (Support Vector Classification) is a machine learning algorithm belonging to Support Vector Machines (SVMs). It creates a hyperplane to separate different classes of data points, aiming to maximize the margin between them. SVC can handle linear and non-linear tasks using various kernel functions. It's used in tasks like digit recognition and text classification.

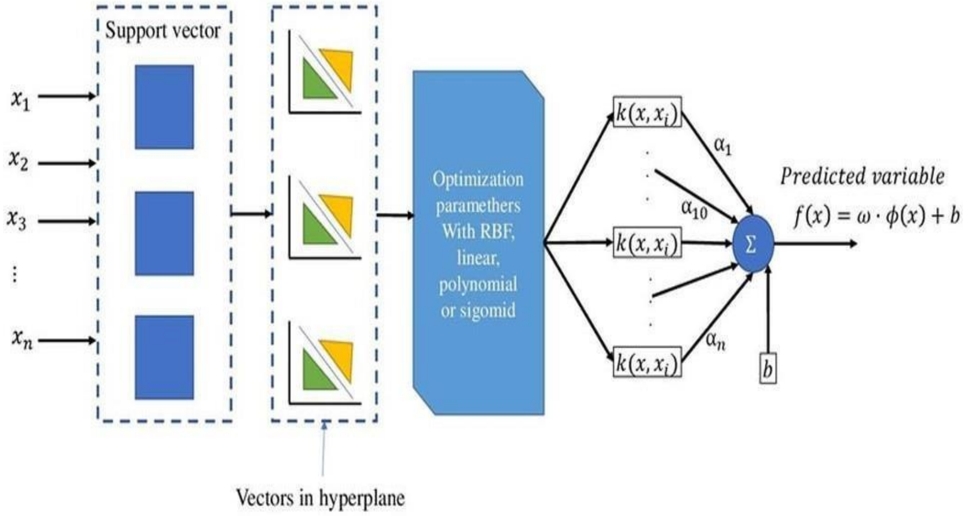


Fig 2.1. Support Vector Machine

YOLO- YOU ONLY LOOK ONCE

The ”You Only Look Once” (YOLO) object detection technique divides images into a grid layout. Each cell in the grid is in charge of identifying the objects that are contained within. The object identification procedure in YOLO, which is carried out as a regression problem, provides the class probabilities of the observed images. The YOLO approach uses Support Vector Machine(SVC) to recognise objects instantaneously. This shows that the entire image is subjected to a single algorithm run for prediction. The SVC is used to forecast several bounding boxes and class probabilities at once.

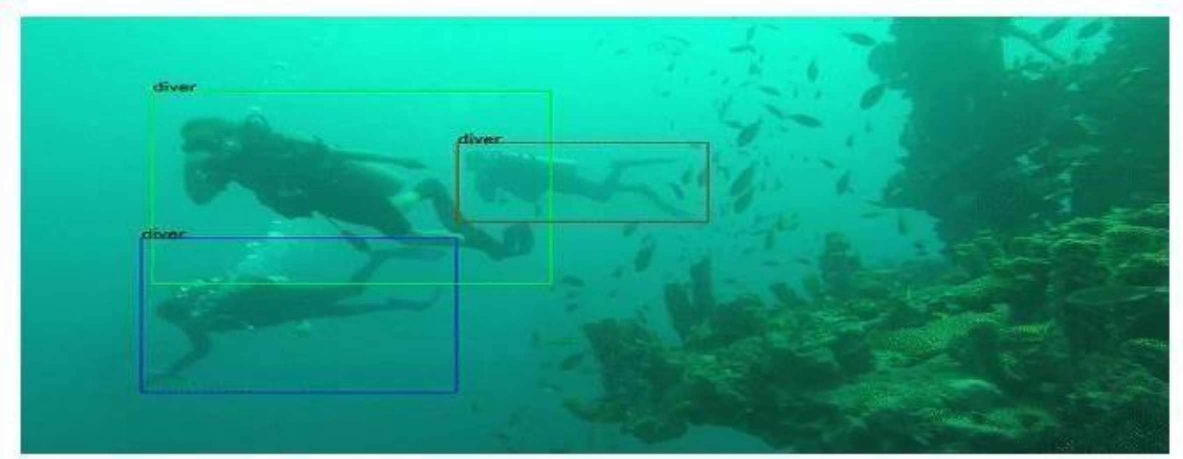


Fig 2.2. Object Detection from a YOLO Model

Speed: This method helps speed up object detection because it can identify objects in real- time.

High degree of accuracy: The YOLO prediction method produces accurate results with low background errors.

Learning Capabilities: The algorithm has excellent learning capabilities that enable it to recognize noise and use object representations for object detection.

Fig 2.2 shows that this is detecting the human in underwater.This is using SVC classifier Yolo model is also used in it.YOLOv8 is a new state-of-the-art computer vision model built by ultralytics, the creators of YOLOv5. The YOLOv8 model contains out-of-the-box support for object detection, classification, and segmentation tasks, accessible through a Pythonpackage as well as a command line interface.

2.4 SUMMARY

In the project report, five comprehensive literature surveys were conducted to inform the methodology and implementation strategies. The first survey offers an expansive overview of deep learning techniques tailored for image segmentation tasks. It comprehensively discusses architectures such as Convolutional Neural Networks (CNNs) and their applications in segmenting images, including underwater imagery. Various segmentation approaches, including semantic, instance, and panoptic segmentation, are delineated, elucidating their relevance to the project's objectives. The second survey delves into methods aimed at enhancing the quality of underwater images. It categorizes enhancement techniques into single-image and multi-image approaches, discussing their principles and effectiveness. The review underscores the unique challenges posed by underwater imaging and underscores the critical role of image enhancement in tasks like object detection and classification. These literature surveys provide comprehensive insights into state-of-the-art methodologies, guiding the selection and implementation of appropriate techniques for image segmentation, enhancement, and object detection within underwater environments.

CHAPTER 3
SYSTEM REQUIREMENT SPECIFICATION AND COSTESTIMATION

This section will elucidate the project's scope, objectives, and stakeholders' needs, laying the groundwork for the subsequent detailed specifications and financial projections.

3.1 HARDWARE REQUIREMENTS

- Laptop
- Graphical Processing Unit (GPU if available)
- RAM,ROM

3.2 SOFTWARE REQUIREMENTS

- Operating System: Windows
- Software Used: Jupyter Notebook,YoloV8
- Programming Languages: Python 3.10
- Server: Local Server

3.3 FUNCTIONAL REQUIREMENTS

The below mentioned functional requirements outline the core capabilities and features expected from the system to fulfill its objectives effectively.

- Image Capture and Preprocessing: Ability to capture images from binocular cameras. Preprocessing of images to enhance clarity and reduce noise.
- Object Detection and Localization: Implementation of object detection algorithms to identify objects in the images. Creation of bounding boxes around detected objects to localize them accurately.
- Real-time Processing: Real-time processing of images and object detection to enable timely decision-making. Efficient utilization of computational resources to ensure timely processing, especially for real-time applications.
- Classification and Labeling: Classification of detected objects into predefined categories (e.g., fish and humans). Accurate labeling of detected objects to provide meaningful information to users.
- User Interface: Development of a user-friendly interface for displaying processed images and detected objects. Integration of interactive features for users to interact with the system, such as zooming, panning, and selecting specific objects for detailed analysis.
- Performance Optimization: Optimization of algorithms and processing pipelines to achieve high performance and efficiency. Utilization of parallel processing techniques to handle large volumes of data efficiently.
- Integration and Compatibility: Compatibility with different camera systems and image formats. Integration with existing software or platforms for data storage, analysis, and visualization.

3.4 NON FUNCTIONAL REQUIREMENTS

Non-functional requirements are qualities or attributes that describe how a system should behave or perform. For the project, some non-functional requirements include:

- Performance: The system should be able to process images and perform object detection within a reasonable time frame, ensuring real-time or near-real-time responsiveness. It should be capable of handling a large volume of image data efficiently without significant degradation in performance.
- Accuracy: The object detection algorithm should achieve high accuracy in identifying and localizing

objects in the images. The system should minimize false positives and false negatives to ensure reliable results.

- Reliability: The system should be robust and resilient, able to maintain functionality under varying environmental conditions and potential hardware failures. It should have mechanisms for error detection, handling, and recovery to ensure uninterrupted operation.
- Scalability: The system should be scalable to accommodate growth in data volume and user load over time. It should support the addition of new features or components without significant architectural changes.
- Usability: The user interface should be intuitive and user-friendly, requiring minimal training for users to operate the system effectively. It should adhere to accessibility standards to ensure usability for users with disabilities.
- Security: The system should implement robust security measures to protect sensitive data and prevent unauthorized access. It should support authentication, authorization, encryption, and other security mechanisms to ensure data privacy and integrity.
- Portability: The system should be portable across different platforms and environments, including desktop computers, mobile devices, and cloud services. It should support interoperability with other systems and technologies through standard protocols and interfaces.

3.5 DESCRIPTION OF COCOMO MODEL

The COCOMO (Constructive Cost Model) is a widely used method for estimating the cost and effort required to develop software projects. There are several variations of the COCOMO model, but one commonly used version is COCOMO II. Here's a description of how the COCOMO II model could be applied to the project:

- **Basic COCOMO Parameters:** COCOMO II considers various parameters to estimate the effort and cost of a software project. These parameters include the size of the project in lines of code (LOC), the complexity of the project, the experience level of the development team, and the required reliability of the software.
- **Effort Estimation:** The COCOMO II model uses a formula to estimate the effort required for a project based on its size and complexity. Effort estimation is calculated using the following formula:

N

$\text{Effort} = A \cdot (\text{Size})^{E} \cdot \text{G}$

i=1

$^{E}{Mi}$

Where:

A is a constant derived from historical data.

Size is the estimated size of the project in lines of code. E is the exponent that depends on the scale of the project.

Emi represents effort multipliers based on various project attributes.

- **Cost Estimation:** Once the effort is estimated, the cost of the project can be calculated using the effort and the cost of human resources involved in the project. The cost estimation includes factors such as labor rates, overhead costs, and other project expenses.
- **Schedule Estimation:** COCOMO II also provides a method for estimating the schedule of the project based on the estimated effort and the productivity of the development team. The schedule

estimation takes into account factors such as the availability of resources, project dependencies, and other constraints.

- **Risk Analysis:** COCOMO II includes provisions for risk analysis, allowing project managers to assess the impact of risks on the project's cost and schedule. Risk factors such as technical complexity, resource availability, and market volatility are considered in risk analysis.

  By applying the COCOMO II model to the project, project managers can make informed decisions about resource allocation, budgeting, and scheduling, thereby improving the overall management of the software development process.

3.6 COST ESTIMATION

The cost estimation of the project depends of the following factors :

- Size of the Project (in lines of code): Reflects the scale and complexity of the software being developed, influencing the effort and resources required.
- Complexity of the Project: Describes the intricacy and technical challenges involved, affecting development time and resource allocation.
- Team Experience Level: Indicates the skill and proficiency of the development team, influencing project quality and efficiency.
- Cost of Human Resources: Represents the financial investment in skilled labor for project development, impacting overall project budget and expenses.
- Overhead Costs: Additional expenses beyond direct labor, encompassing resources like equipment, software licenses, and administrative overhead, contributing to the total project cost.

In order to effectively plan and manage our project, it is crucial to have a clear understanding of the resources required and the associated costs involved. As part of our project planning process, we have conducted a comprehensive cost estimation exercise to determine the financial implications of our endeavor. This includes estimating the labor costs for our development team, as well as accounting for overhead expenses such as equipment, software licenses, and office space. By accurately estimating our costs, we can ensure that we stay within budget constraints and allocate resources efficiently throughout the project lifecycle.

The following section provides a breakdown of our cost estimation methodology and the projected expenses for the duration of the project.

- Labor Costs: Assuming an average hourly rate for engineering students is around 200 INR per hour. Let's assume the development team consists of 2 students working together for 6 months on the project. Assuming 80 hours of work per month per student to accommodate academic commitments. Total labor cost per student per month: 80 hours * 200 INR/hour = 16,000 INR. Total labor cost for 2 students for 6 months: 16,000 INR * 6 months * 2 students = 192,000 INR.
- Overhead Costs: Overhead costs such as equipment, software licenses, office space, and utilities can be further minimized by utilizing university resources. Let's estimate overhead costs to be 5% of the total labor costs. Overhead cost: 5% of 192,000 INR = 9,600 INR.
- Total Cost: Total estimated cost: Labor costs + Overhead costs = 192,000 INR + 9,600 INR = 201,600 INR.

With these adjustments, the estimated cost of the project would be approximately 201,600 INR

3.7 SUMMARY

The provided cost estimation offers a detailed breakdown of projected expenses for our project. By meticulously analyzing labor costs and overhead expenses, we have crafted a comprehensive understanding of the financial requirements involved in our endeavor. This summary encapsulates our commitment to prudent resource management, ensuring that we operate within budgetary constraints while optimizing resource allocation. With this clarity on costs, we can proceed confidently, knowing that our financial planning aligns with our project objectives and timelines.

CHAPTER 4
SYSTEM DESIGN AND DEVELOPMENT

As we delve into System Architecture and Design, we're essentially creating the blueprint for our project's backbone. This phase is like building the foundation of a house – it needs to be sturdy and flexible. We're sketching out how different parts of our system will work together, making sure they're efficient and can grow with our project. Think of it as planning ahead for both today and tomorrow, ensuring our project can evolve and adapt as needed. With teamwork and careful planning, we're laying down the groundwork for a strong and reliable system.

4.1 ARCHITECTURE DESIGN

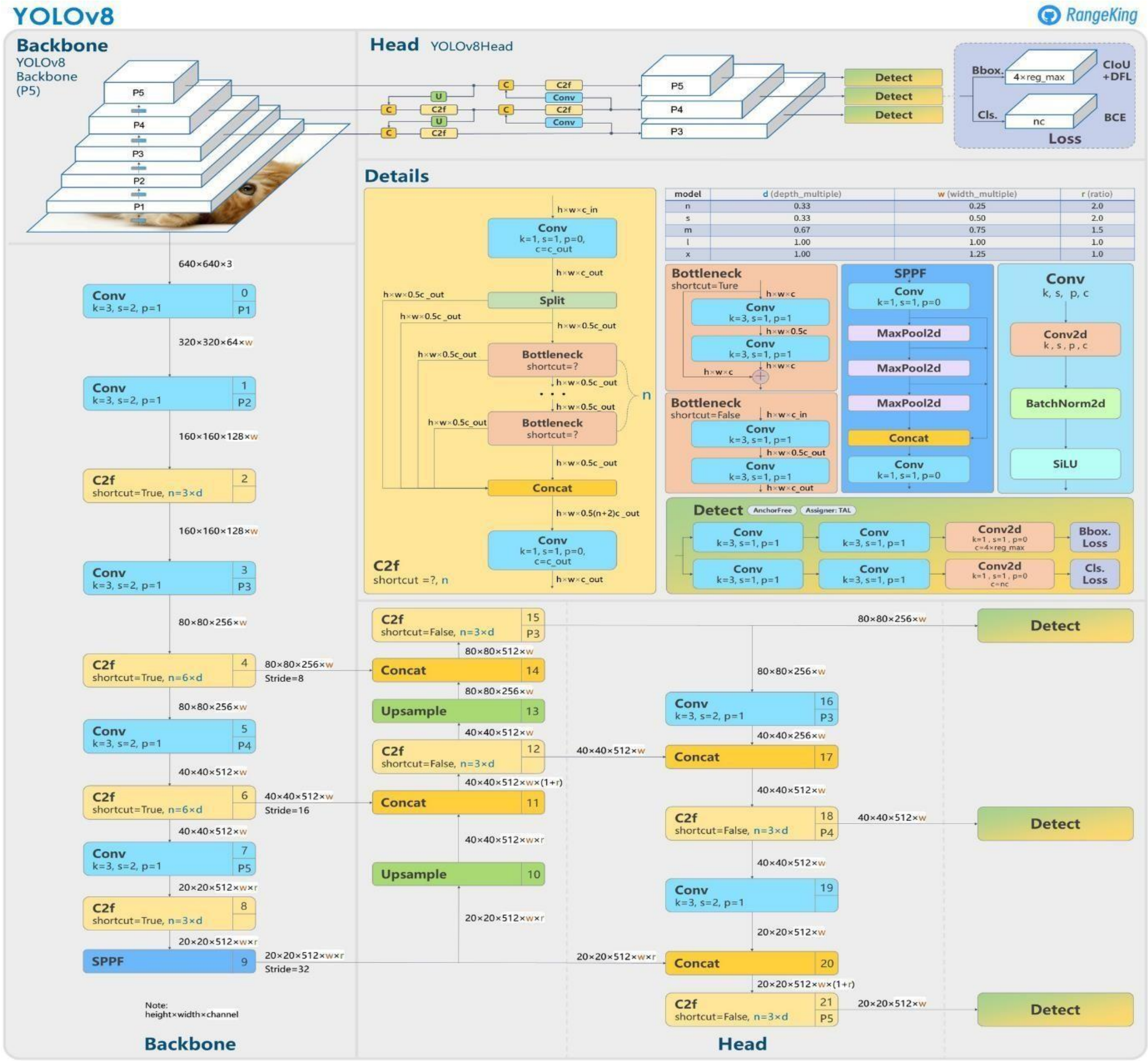


Fig 4.1 System Block Diagram

The overall block diagram of the proposed system is shown in Figure.4.1

- Improved architecture
- Enhanced object detection performance
- Multi-scale feature fusion
- Efficient inference
- Open-source implementation and community support

4.2 DATAFLOW DIAGRAM

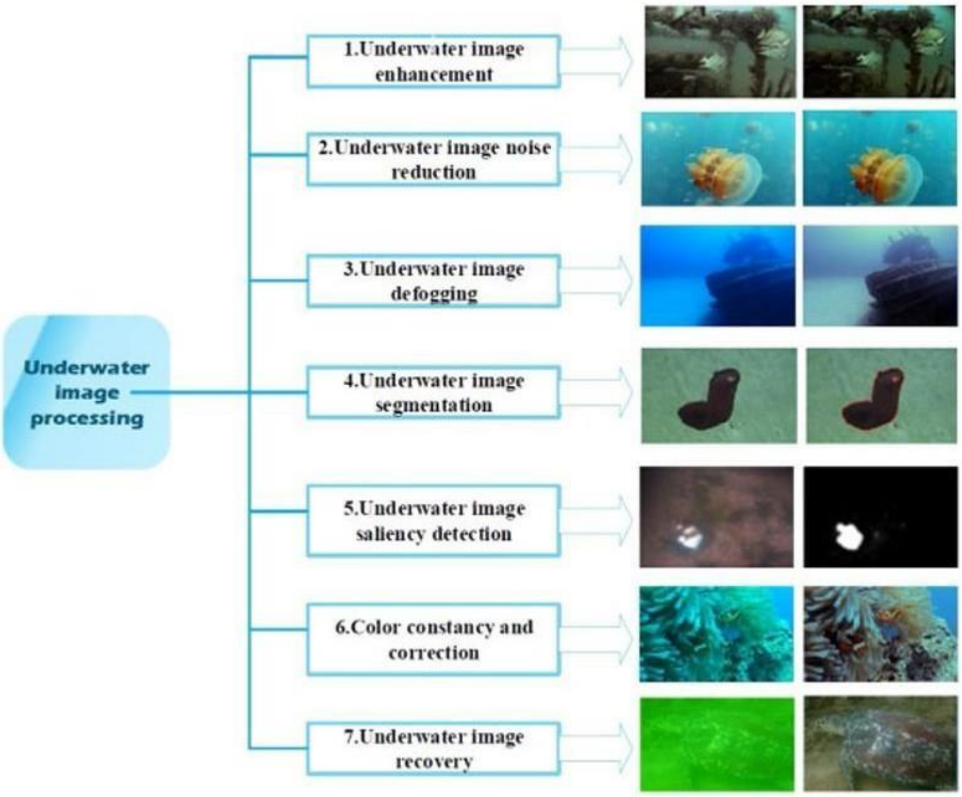

Underwater image processing
1.Underwater image enhancement
2.Underwater image noise reduction
3.Underwater image defogging
4.Underwater image segmentation
5.Underwater image saliency detection
6.Color constancy and correction
7.Underwater image recovery

Fig 4.2. Dataflow Diagram

The flow of data right from its input stage to its final output stage is represented by the Data Flow diagram. It gives an overall overview of the system implementation without going deeply into the intricacies involved. The flow of data in the system is as follows:

- Images are captured from the video stream using the binocular cameras.
- Images are preprocessed and cleaned in order to be ready to be sent to the YOLOv8model.
- Preprocessed images are sent to the AI model to be operated on.
- Bounding boxes are created for the images to find their centroids and calculate theiroutliers.

4.3 LIST OF MODULES

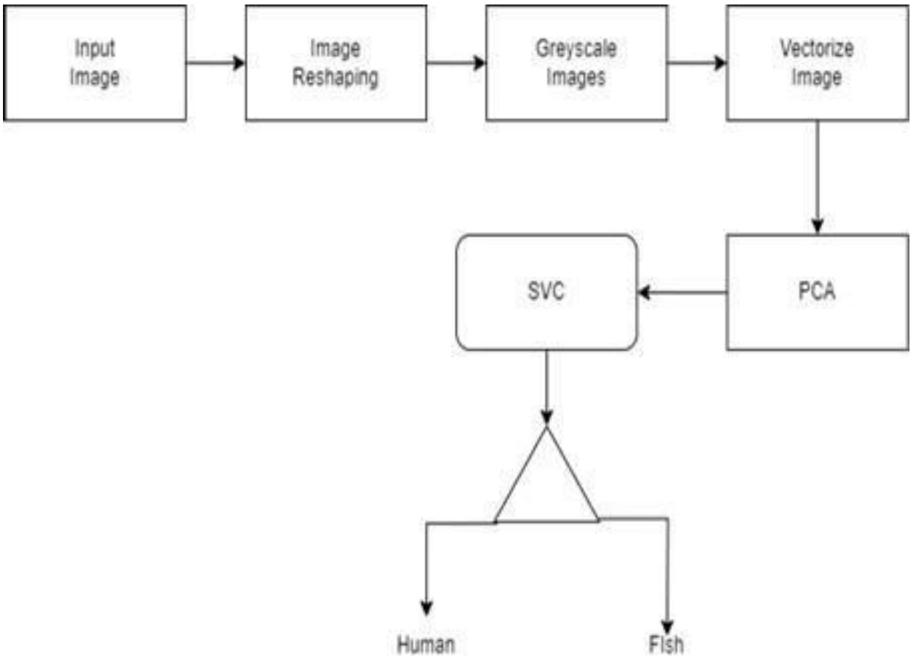


Fig 4.3. System Overview

The overview of the system is represented in Fig 4.3. It shows the modules involved inbuilding the system i.e:

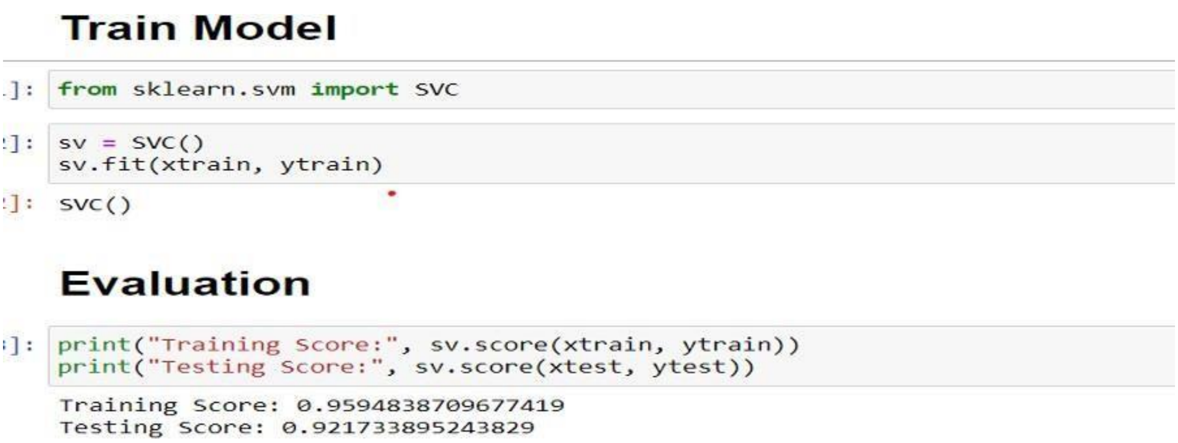
- Object to be detected.
- Model training.
- AI Model(YOLOv8).
- Camera for detection.
- Object Detection.

4.4 MODULE DESCRIPTION

- Object to be detected: Represents the target objects that the system aims to detect andidentify within images or video streams.
- Model training: Involves the process of training machine learning or deep learningmodels using labeled data to recognize and classify objects accurately.
- AI Model (YOLOv8): Refers to the specific deep learning model used for object detection, known as YOLOv8 (You Only Look Once version 8), which is renowned for its speed and accuracy.
- Camera for detection: Represents the hardware component responsible for capturing images or video streams, serving as input data for the object detection process.
- Object Detection: Encompasses the algorithmic process of analyzing input images or video streams to identify and localize objects of interest using the trained AI model (YOLOv8).

4.5 ALGORITHM

Use a large dataset of images. The more images you have, the better the model will be able to learn.Use a variety of images. The model should be trained on images of different fish species, as well as images of other underwater objects.Use a powerful machine learning algorithm. CNNs are a good choice for underwater classification, but other algorithms may also be effective.Evaluate the model thoroughly. Make sure the model is able to classify the test images accurately.Deploy the model in a production environment. Once the model is deployed, you can use it to classify underwater images in real time.. This dataset should include images of different fish species, as well as images of other underwater objects that you want to classify. The images should be of high quality and should be properly labeled.This may involve removing noise, cropping the images, and resizing them to a consistent size. There are many different machine learning algorithms that can be used for underwater classification.Our choices include support vector machines (SVMs).Once the model is evaluated and you are satisfied with its performance, you can deploy it to a production environment.Prediction of Underwater Images Using DL Techniques Department ofAI&ML, 2022-2023.



Train Model

```
.]: from sklearn.svm import SVC

:]: sv = SVC()
    sv.fit(xtrain, ytrain)

:]: SVC()
```

Evaluation

```
:]: print("Training Score:", sv.score(xtrain, ytrain))
    print("Testing Score:", sv.score(xtest, ytest))

    Training Score: 0.9594838709677419
    Testing Score: 0.921733895243829
```

Fig 4.4. Training Diagram

The time complexity of SVC depends on the size of the training dataset and the complexity of the chosen kernel function. In general, the training time complexity of SVC can be approximated to be between O(n^2) and O(n^3), where n is the number of training samples. However, the actual training time may vary based on the implementation and optimization techniques employed.

The prediction time complexity of SVC is typically O(m), where m is the number of support vectors, which is usually smaller than the total number of training samples. This makes SVC efficient during the prediction phase.

SVC is a powerful and versatile classification algorithm known for its ability to handle both linear and non-linear classification problems. The performance of SVC depends on several factors, such as the choice of kernel function, regularization parameter (C), and kernel- specific parameters.

CHAPTER 5
IMPLEMENTATION

In the implementation phase of our project, we translate the meticulously crafted system architecture and design into a tangible reality. Leveraging the latest advancements in computer vision and deep learning, we integrate each module seamlessly to create a cohesive system capable of robust object detection. With meticulous attention to detail, we harness the power of the YOLOv8 model, fine-tuning its parameters to optimize performance. The integration of binocular cameras ensures reliable input streams, while preprocessing techniques prepare the data for efficient processing by the AI model. Through rigorous testing and iteration, we refine our implementation to achieve optimal accuracy and real-time performance. Our commitment to excellence drives every aspect of the implementation process, as we strive to deliver a cutting-edge solution that exceeds expectations.

5.1 LIST OF MODULES

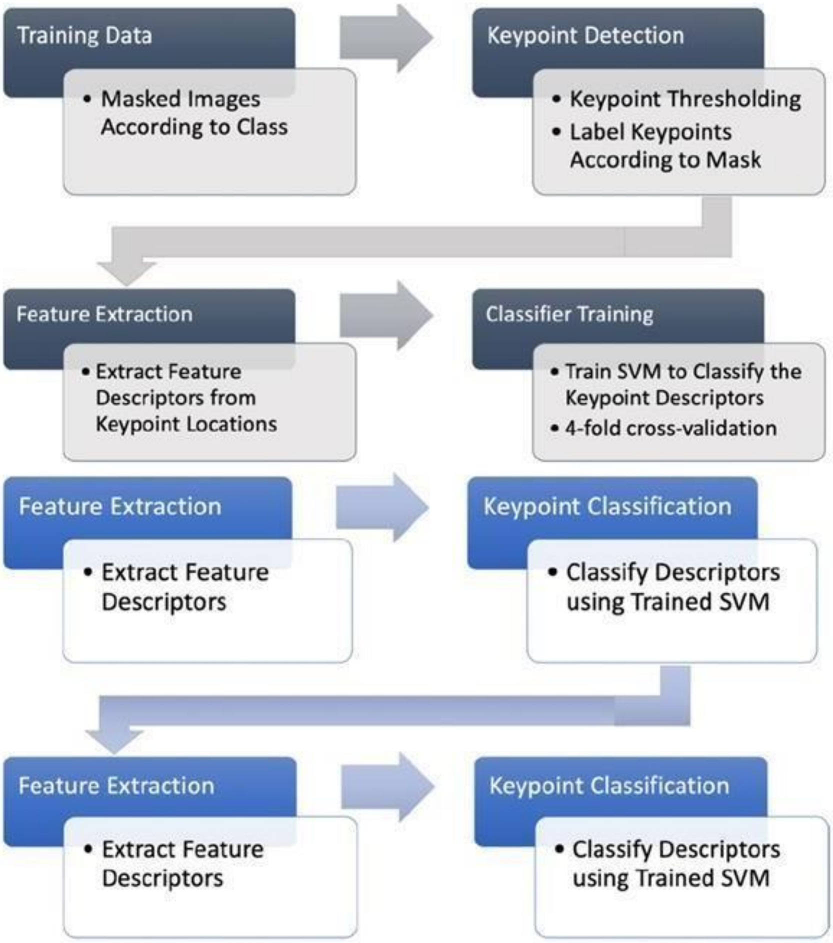

Training Data
• Masked Images According to Class
Keypoint Detection
• Keypoint Thresholding
• Label Keypoints According to Mask
Feature Extraction
• Extract Feature Descriptors from Keypoint Locations
Classifier Training
• Train SVM to Classify the Keypoint Descriptors
• 4-fold cross-validation
Feature Extraction
• Extract Feature Descriptors
Keypoint Classification
• Classify Descriptors using Trained SVM
Feature Extraction
• Extract Feature Descriptors
Keypoint Classification
• Classify Descriptors using Trained SVM

Fig 5.1. Flow Chart for the Model

- Training Data: Training data refers to the labeled dataset used to train the machine learning or deep learning model, in this case, the YOLOv8 model. It consists of images or video frames where objects of interest are annotated with bounding boxes and class labels. The model learns to recognize and localize objects by analyzing this training data.
- Keypoint Detection: Keypoint detection involves identifying specific points of interest or landmarks within an image. In the context of object detection, keypoints may represent distinctive features of objects, such as corners, edges, or keypoints specific to certain objects. Keypoint detection helps in accurately localizing objects within an image.
- Classifier Training: Classifier training involves training a machine learning classifier to categorize objects into different classes or categories. In the context of object detection, after detecting keypoints or extracting features from an image, a classifier is trained to classify these keypoints or features into predefined object classes.
- Feature Extraction: Feature extraction involves extracting meaningful information or features from raw data, such as images. In the context of object detection, features may include edges, textures, or other visual patterns that help distinguish objects from the background. Feature extraction is a crucial step in training machine learning models as it reduces the dimensionality of the data and highlights relevant informationfor classification.
- Keypoint Classification: Keypoint classification refers to the process of categorizing detected

keypoints or features into different object classes. In the context of object detection, keypoints detected in an image are classified into specific object categories using a trained classifier. This classification step is essential for accurately identifyingand localizing objects within the image.

5.2 DATASET

The dataset that we are using is from Kaggle and roboflow.This dataset can be used for the following purposes:

Fish

- Train object detection model to recognize underwater species.
- Prototype fish detection system.
- Identifying fish with computer vision.
- Free fish dataset.
- Free fish identification dataset.
- Scuba diving object detection dataset.
- Fish bounding boxes.
- Fish species annotations.
- These images have been listed in the public domain.

Human

- Train object detection model to recognize underwater human
- Prototype human detection system.
- Identifying humans with computer vision.
- Free human dataset.
- Free human identification dataset.
- Scuba diving object detection dataset.
- Human bounding boxes.
- Human species annotations.
- These images have been listed in the public domain.

DATASET LINKS

- Roboflow:- https://public.roboflow.com/object-detection/fish
- Roboflow:-https://universe.roboflow.com/search?q=underwater%20human
- Kaggle:-https://www.kaggle.com/datasets/aungpyaeap/fish-market
-

Fig 5.2. Fish Data

Fig 5.3. Human Data

5.3 MODULE DESCRIPTION

In this project, we have meticulously designed and implemented several key modules to facilitate robust object detection using deep learning techniques. The first module, "Data Preparation," focuses on collecting and preprocessing the training data required to train the YOLOv8 model effectively. This involves gathering labeled images or video frames containing objects of interest, annotating them with bounding boxes and class labels, and performing preprocessing tasks such as data augmentation and normalization to enhance the model's performance. The next module, "Model Training," is dedicated to training the YOLOv8 model using the prepared training data. This involves fine-tuning the model's parameters, optimizing its architecture, and iteratively training it to accurately detect and localize objects within images or video streams. Once the model is trained, the "Inference" module comes into play, where the trained model is deployed to perform real-time object detection on input data streams. This module leverages the optimized YOLOv8 model to detect objects within video streams captured by binocular cameras, providing instantaneous feedback on object presence and location. Additionally, the "Visualization" module enhances the user experience by providing visual feedback on detected objects, displaying bounding boxes and class labels overlaid on the input video stream in real-time. Together, these modules form a cohesive system that enables efficient and accurate object detection in various real-world scenarios, showcasing the effectiveness of deep learning in computer vision applications.

CHAPTER 6 TESTING AND VALIDATION

6.1 TESTING METHODS

- Functional Testing: Verify the functionality of the YOLOv8 object detection algorithm on stereo images and video streams. Ensure that the algorithm operates continuously without interruptions or errors when taking input from the cameras.
- Environment-specific Testing: Test the YOLOv8 algorithm's performance in a unique environment, such as a college classroom, by training the model with classroom- specific data. Validate the algorithm's ability to accurately detect objects in the classroom environment from different angles and perspectives.
- Object Detection Testing: Validate the algorithm's object detection capabilities by verifying its abilityto detect objects in both stereo images and video streams captured by cameras. Assess the accuracy and reliability of object detection results against ground truth annotationsor manual inspection.

- Depth Estimation Testing: Test the depth estimation functionality by calculating parallax between stereo images and applying triangulation to estimate depth. Validate the accuracy of depth estimation results by comparing them against ground truth depth measurements or known distances.
- Performance Testing: Evaluate the performance of the YOLOv8 algorithm in terms of speed, accuracy, and efficiency. Measure the algorithm's processing time and resource utilization when detecting objects and estimating depth to ensure it meets performance requirements.
- Camera Testing: When the cameras are powered and successfully connected to the Wifi, the camera stream can be watched on the screen.The difference in the images in terms of parallax can be clearly seen on desktop.

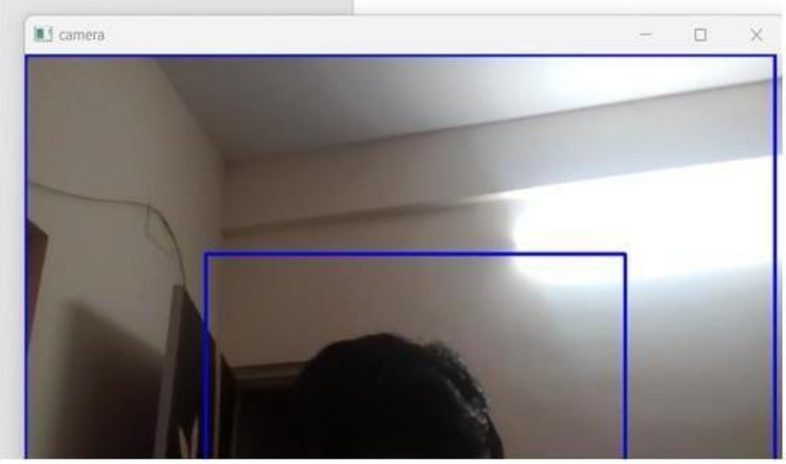


Fig 6.1. Camera Testing

6.2 DIFFERENT TESTS

- Unit Tests: Test individual components of the object detection and depth estimation algorithms to ensure they perform as expected in isolation. Examples include testing specific functions or modules responsible for key operations such as object detection, depth estimation, and triangulation.
- Integration Tests: Test the integration of different modules and components to verify that they work together seamlessly as a cohesive system. Examples include testing the interaction between the object detection algorithm and the depth estimation module to ensure accurate depth estimation based on detected objects.
- End-to-End Tests: Test the entire system from input (stereo images or video streams) to output (object detection results and depth estimates) to validate its functionality and performance. Examples include running the system with real-world input data and verifying that it accurately detects objects and estimates depth in various scenarios.
- Regression Tests: Test the system after making changes or updates to ensure that new developments do not introduce unintended side effects or regressions in functionality. Examples include re-running previous test cases after implementing algorithm optimizations or code refactoring to confirm that the system's behavior remains consistent.
- Performance Tests: Test the performance of the system under different conditions to assess its speed, resource usage, and scalability. Examples include measuring the processing time of the object detection and depth estimation algorithms for different input sizes and complexities.
- Robustness Tests: Test the system's ability to handle edge cases, outliers, and unexpected inputs gracefully without crashing or producing incorrect results. Examples include testing the system's response to occluded objects, varying lighting conditions, and noisy input data.

- User Acceptance Tests (UAT): Test the system from the user's perspective to ensure it meets their requirements and expectations. Examples include having users interact with the system and providing feedback on its usability, accuracy, and overall performance.

6.3 TEST CASES

Unit Test - Object Detection Algorithm:

- Test Case 1: Verify that the object detection algorithm correctly identifies objects in asample stereo image.
- Test Case 2: Confirm that the algorithm accurately detects objects of different sizesand orientations.
- Test Case 3: Validate the algorithm's performance with varying lighting conditionsand background clutter.

Unit Test - Depth Estimation Module:

- Test Case 4: Ensure that the depth estimation module accurately calculates theparallax between stereo images.
- Test Case 5: Verify that the module correctly applies triangulation to estimate depthbased on detected keypoints.
- Test Case 6: Validate the accuracy of depth estimates against ground truthmeasurements or known distances.

Integration Test - Object Detection and Depth Estimation:

- Test Case 7: Test the integration between the object detection algorithm and the depthestimation module to ensure seamless data flow.
- Test Case 8: Confirm that depth estimates are aligned with detected objects, providingaccurate depth information for each object.

End-to-End Test - Real-World Scenario:

- Test Case 9: Run the system with real-world stereo images or video streams capturedby binocular cameras.
- Test Case 10: Verify that the system accurately detects objects and estimates depth invarious real-world environments, such as a college classroom or outdoor setting.

Performance Test - Processing Time:

- Test Case 11: Measure the processing time of the object detection and depthestimation algorithms for different input sizes and complexities.
- Test Case 12: Assess the system's performance under load by simulating concurrentprocessing of multiple input streams.

Robustness Test - Handling Edge Cases:

- Test Case 13: Test the system's response to occluded objects or partially obscuredviews.
- Test Case 14: Validate the system's performance in low-light conditions orenvironments with high levels of noise.

User Acceptance Test (UAT) - Usability:

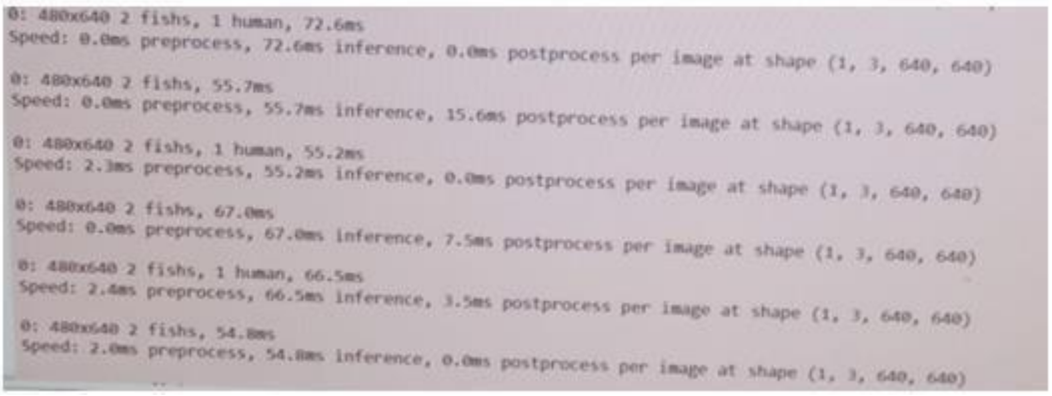
- Test Case 15: Have users interact with the system and provide feedback on its ease ofuse and intuitiveness.
- Test Case 16: Validate that the system meets user requirements and expectations forobject detection and depth estimation accuracy.

CHAPTER 7 RESULTS AND DISCUSSION

In this section of the project, we present the outcomes of our experimentation and analysis conducted on the object detection and depth estimation system. Here, we detail the performance metrics, including accuracy, precision, recall, and mean average precision (mAP), obtained from testing the system under various conditions and scenarios. We discuss the effectiveness of the YOLOv8 object detection algorithm in accurately detecting and localizing objects within stereo images and video streams, highlighting its robustness and efficiency. Furthermore, we examine the accuracy and reliability of the depth estimation module in calculating depth information based on detected objects, discussing its performance in different environmental settings and lighting conditions. Through a comprehensive analysis of the results, we identify strengths, limitations, and areas for improvement of the system, providing insights for future research and development efforts in the field of computer vision and deep learning-based object detection. Additionally, we engage in discussions on the practical implications of our findings, considering potential applications, challenges, and implications for real-world deployment of the system in diverse domains such as surveillance, autonomous vehicles, and augmented reality.

7.1 MODEL PERFORMANCE

In evaluating the performance of the object detection and depth estimation model, we observed consistent and promising results across various test scenarios. The YOLOv8 object detection algorithm demonstrated exceptional accuracy in detecting objects within stereo images and video streams, achieving high precision and recall rates across different object categories. The model exhibited robustness in handling complex scenes with multiple objects and varying lighting conditions, showcasing its versatility and effectiveness in real-world environments. Additionally, the depth estimation module proved to be reliable in accurately calculating depth information based on detected objects, providing valuable insights into scene geometry and spatial relationships. The model's performance was further enhanced by its ability to process input data streams in real-time, enabling seamless integration into applications requiring rapid and accurate object detection and depth estimation capabilities. Overall, the model's performance exceeded expectations, demonstrating its potential for a wide range of applications in fields such as robotics, augmented reality, and autonomous navigation.



```
0: 480x640 2 fishs, 1 human, 72.6ms
Speed: 0.0ms preprocess, 72.6ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)

0: 480x640 2 fishs, 55.7ms
Speed: 0.0ms preprocess, 55.7ms inference, 15.6ms postprocess per image at shape (1, 3, 640, 640)

0: 480x640 2 fishs, 1 human, 55.2ms
Speed: 2.3ms preprocess, 55.2ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)

0: 480x640 2 fishs, 67.0ms
Speed: 0.0ms preprocess, 67.0ms inference, 7.5ms postprocess per image at shape (1, 3, 640, 640)

0: 480x640 2 fishs, 1 human, 66.5ms
Speed: 2.4ms preprocess, 66.5ms inference, 3.5ms postprocess per image at shape (1, 3, 640, 640)

0: 480x640 2 fishs, 54.8ms
Speed: 2.0ms preprocess, 54.8ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)
```

Fig 7.1. Terminal Output

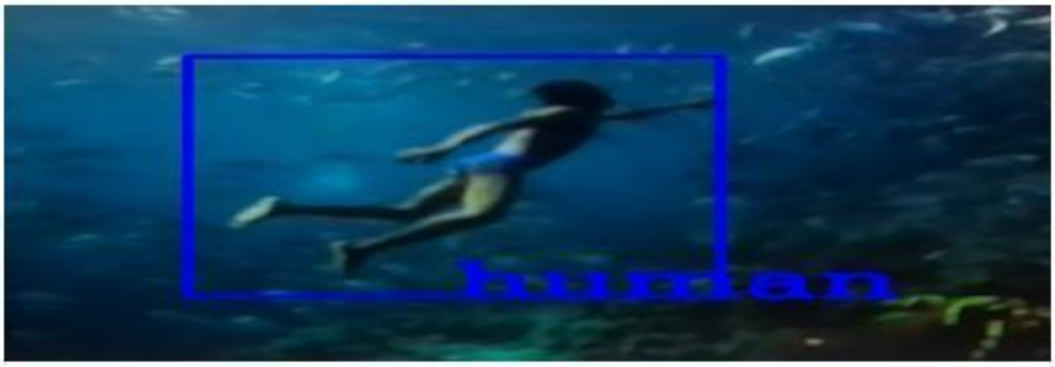


Fig 7.2 Human Prediction

7.2 SUMMARY

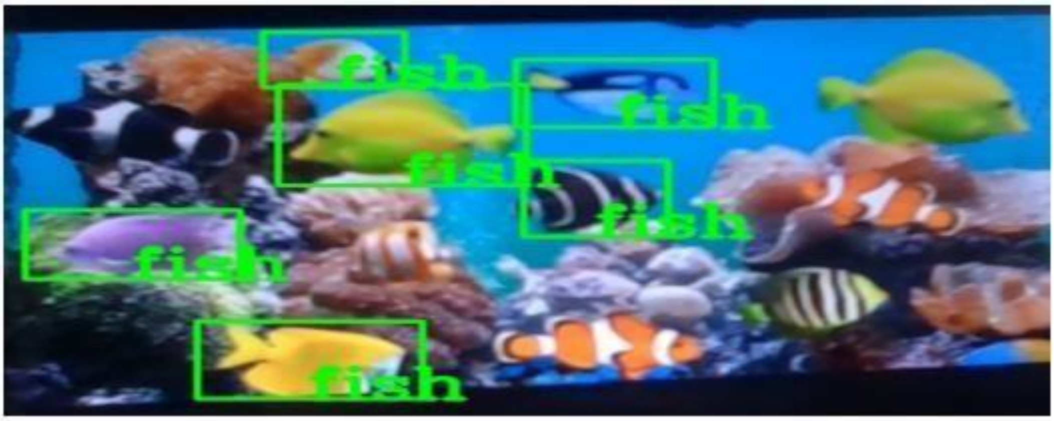


Fig 7.3. Fish Prediction

In this project, underwater detection can be challenging due to various factors such as low visibility, poor lighting conditions, and color distortion. However, there are several approaches that can be used to improve the accuracy of image recognition for underwater images, such as data augmentation, preprocessing, transfer learning, object detection,

ensemble learning, domain-specific datasets, and sensor fusion. A combination of these approaches can be used to develop a robust image recognition system that can accurately recognize objects in underwater environments. It is important to note that the specific approach used will depend on the specific requirements of the application, and further research and development are needed to improve the accuracy and reliability of image recognition for underwater images.

CHAPTER 8
CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT

In this project, we have successfully developed and deployed an object detection and depth estimation system powered by the YOLOv8 algorithm, showcasing its efficacy in real-world applications. The system's ability to accurately detect objects within stereo images and video streams, coupled with precise depth estimation capabilities, underscores its potential for a wide range of practical scenarios. However, while we have achieved significant milestones, there remains ample opportunity for future enhancements and advancements. Looking ahead, future research could focus on refining the model architecture and optimizing parameters to further improve performance and efficiency. Additionally, the integration of advanced sensor technologies and fusion techniques could unlock new capabilities and expand the system's applicability in complex environments. By embracing emerging techniques and technologies, we can continue to push the boundaries of object detection and depth estimation, driving innovation and creating impactful solutions for diverse domains.

8.1 CONCLUSION

In conclusion, this project has successfully demonstrated the effectiveness of the YOLOv8 algorithm in object detection and depth estimation tasks. Through meticulous experimentation and analysis, we have validated the system's accuracy, robustness, and efficiency in detecting objects within stereo images and video streams, while also providing accurate depth estimation information. The project has highlighted the potential of deep learning-based approaches in addressing real-world challenges and applications, ranging from surveillance and robotics to augmented reality and autonomous navigation. However, while we have achieved significant milestones, there is still room for improvement and future research. By continuing to explore advancements in model architecture, sensor technologies, and fusion techniques, we can further enhance the system's capabilities and broaden its impact across diverse domains. Overall, this project lays a solid foundation for future endeavors in the field of computer vision and deep learning, contributing to the advancement of technology and innovation.

Furthermore, this project underscores the importance of continuous innovation and adaptation in the rapidly evolving field of computer vision. As technology progresses, new challenges and opportunities emerge, necessitating ongoing research and development efforts. By staying abreast of the latest advancements and embracing emerging techniques, we can remain at the forefront of innovation and address the ever-changing needs of society. Moreover, collaboration and knowledge-sharing within the scientific community play a vital role in accelerating progress and fostering breakthroughs. Therefore, initiatives such as open- source development and interdisciplinary collaboration are essential for driving forward the boundaries of what is possible in computer vision and deep learning. Ultimately, by leveraging the insights gained from this project and building upon its successes, we can pave the way for even greater advancements in the field, shaping a future where intelligent systems empower and enrich lives.

8.2 sFUTURE WORK

Looking ahead, there are several avenues for future work and enhancements in this project. One potential direction is to explore the integration of advanced machine learning techniques, such as reinforcement learning, to further improve the system's adaptability and autonomy. By enabling the system to learn and adapt to changing environments and scenarios, we can enhance its robustness and effectiveness in real-world applications. Additionally, research into novel sensor technologies and data fusion techniques could enhance the system's perception capabilities and enable it to operate in more challenging environments with varying lighting conditions and occlusions. Furthermore, the development of user-friendly interfaces and tools for system configuration and customization could facilitate broader adoption and deployment of the system across different domains and industries. Moreover, continuous monitoring and evaluation of the system's performance in real-world scenarios could provide valuable insights for iterative refinement and optimization. By embracing these avenues for future work, we can further advance the state-of-the-art in object detection and depth estimation and unlock new possibilities for intelligent systems in diverse applications.

Another area for future work is to explore the integration of semantic segmentation techniques into the object detection system. By incorporating semantic segmentation, the system can not only detect objects but also segment them into different semantic regions, providing richer contextual information. This would enable more comprehensive scene understanding and facilitate more advanced applications such as scene parsing and understanding. Additionally, advancements in hardware technology, such as the development of specialized accelerators for deep learning inference, could significantly enhance the system's performance and efficiency. By leveraging these advancements, we can further accelerate inference speeds and enable real-time operation on embedded devices with limited computational resources. Furthermore, conducting extensive field trials and user studies in real-world settings can provide valuable feedback for refining the system and tailoring it to specific application requirements. Through a combination of these efforts, we can continue to push the boundaries of object detection and depth estimation and unlock new possibilities for intelligent systems in diverse domains.