UNDERWATER OBJECT DETECTION USING YOLO

TEJASWINI SINGH –RA2211026010175
HARSHITHA MEENAKSHI A–RA2211026010178
MISHRIYAA VILLURI-RA2211026010184

INTRODUCTION

Monitoring underwater ecosystems is essential for marine research, but traditional methods are time-consuming and inefficient. Recent advances in computer vision, particularly the YOLO (You Only Look Once) model, have made real-time object detection more feasible in challenging environments like underwater. Our project, "Underwater Object Detection Using YOLO," aims to develop an automated system to detect and classify underwater animals efficiently. This system will aid researchers by providing real-time, accurate detection, addressing challenges like murky water and limited datasets, and helping to enhance marine conservation efforts.

WHY THIS PROJECT?

Traditional methods for monitoring underwater ecosystems are time-consuming and inefficient.Advances in computer vision, especially YOLO (You Only Look Once), make real-time underwater object detection more feasible.The system will help researchers and marine biologists by providing accurate, real-time detection in challenging conditions like murky water and limited datasets.It contributes to marine conservation efforts by improving monitoring and research efficiency.

HISTORY OF UNDERWATER OBJECT DETECTION

Early underwater object detection relied on manual observation and sonar technology.With advancements in machine learning and deep learning, models like YOLO have revolutionized object detection.Previous methods struggled with low visibility, poor lighting, and the vast variety of marine species.YOLO-based models now allow faster, more reliable classification of underwater species.

ADVANTAGES

Real-time detection: Enables researchers to monitor ecosystems instantly.Higher accuracy:

Improves classification of underwater animals even in murky water.Automation:

Reduces manual labor and saves time for researchers.Conservation efforts:

Aids in tracking species and studying ecological changes efficientl

CHALLENGES

Murky water & poor lighting: Can affect model accuracy.Limited datasets: Training requires large, well-annotated underwater image datasets.Complex marine environments: Variability in species appearance and movement makes detection difficult.Hardware constraints: Running deep learning models in real-time requires high computational power.

User stories

User Stories: Product Owner

User Story 1: Real-time Underwater Object DetectionAs a Product Owner,

I want the system to detect and classify underwater objects in real-time so that researchers can efficiently study marine life.Acceptance Criteria

:The system processes underwater images and identifies objects in real time.Detection accuracy meets at least 90% of expected results.The model adapts to murky water conditions and various lighting scenarios.

User Story 2: Feedback and Model Improvement

As a Product Owner, I want the system to collect feedback on detection accuracy so that improvements can be made.Acceptance Criteria:Users can report false detections or suggest improvements.The system logs feedback and incorporates model retraining periodically.Accuracy improves with every model update.

User Story 3: Dataset ExpansionAs a Product Owner,

I want the system to allow dataset expansion by integrating new underwater images, ensuring better object classification.Acceptance Criteria:Users can upload new labeled underwater images.The model retrains and incorporates new data into object recognition.The dataset grows over time, increasing detection accuracy.

User stories : Scrum master

1. Role: Project ManagerGoal: Ensure efficient project execution within time, budget, and scope.Acceptance Criteria:A detailed project plan with milestones is shared with stakeholders.Project updates are provided at each sprint review.

2. Role: Data AnalystGoal: Process and analyze underwater datasets for better detection performance.Acceptance Criteria:Dataset is cleaned and prepared before training.The model achieves progressive accuracy improvements after each iteration.

3. Role: Quality Assurance AnalystGoal: Ensure the model performs accurately in different underwater conditions.Acceptance Criteria:Test cases simulate various underwater scenarios (murky water, lighting changes, movement).System meets accuracy benchmarks before deployment.

User stories: Member

User Story 1: Marine ResearcherAs a Marine Researcher, I want the system to help identify species so that I can study marine biodiversity efficiently.Acceptance Criteria:The system detects and classifies at least 90% of marine species correctly.Users can download detection reports for analysis

.User Story 2: DeveloperAs a Developer, I want to collaborate on model improvements and bug fixes to maintain accuracy.Acceptance Criteria:Code is reviewed before merging into production.Model performance is tested after every update

.User Story 3: Environmental Organization StakeholderAs a Stakeholder, I want the system to support conservation efforts by monitoring endangered species.Acceptance Criteria:The system flags rare or endangered marine species.Reports can be exported for further ecological studies.

PRODUCT VISION

1. AudiencePrimary Audience:Marine researchers and environmentalists who need a reliable system for detecting and classifying underwater objects.Oceanographers and conservationists aiming to study marine biodiversity and monitor endangered species.Secondary

2. Audience:Defense and security teams needing underwater object detection for surveillance.Underwater archaeologists analyzing submerged artifacts and structures.Fishermen and divers looking for species identification assistance.

3. NeedsPrimary Needs:Accurate object detection in varying underwater conditions (murky water, different light levels).Real-time processing to support active exploration and research.Robust dataset expansion, allowing the model to adapt to new objects and species.Secondary Needs:Feedback mechanism for users to correct misclassifications and refine the model.Offline functionality for use in remote underwater locations.Compatibility with underwater cameras and drones for seamless deployment.

3. Products
Core Product:A YOLO-based underwater object detection system that identifies and classifies marine life and objects efficiently.Additional Features:Improved UI: User-friendly interface for easier navigation and analysis.Feedback System: Users can mark incorrect detections to enhance future performance.Model Adaptability: Continuous learning from new images and user feedback.—

-4. ValuesCore Values:Efficiency: The system processes images quickly, even in challenging conditions.Accuracy: Designed to minimize false detections and improve classification precision.Scalability: Can be adapted to various underwater exploration needs, from research to security.

Differentiators:Customizable detection models for different user needs (e.g., marine biology vs. security).AI-powered adaptation based on environmental conditions.Integration with real-world underwater devices for seamless functionality.

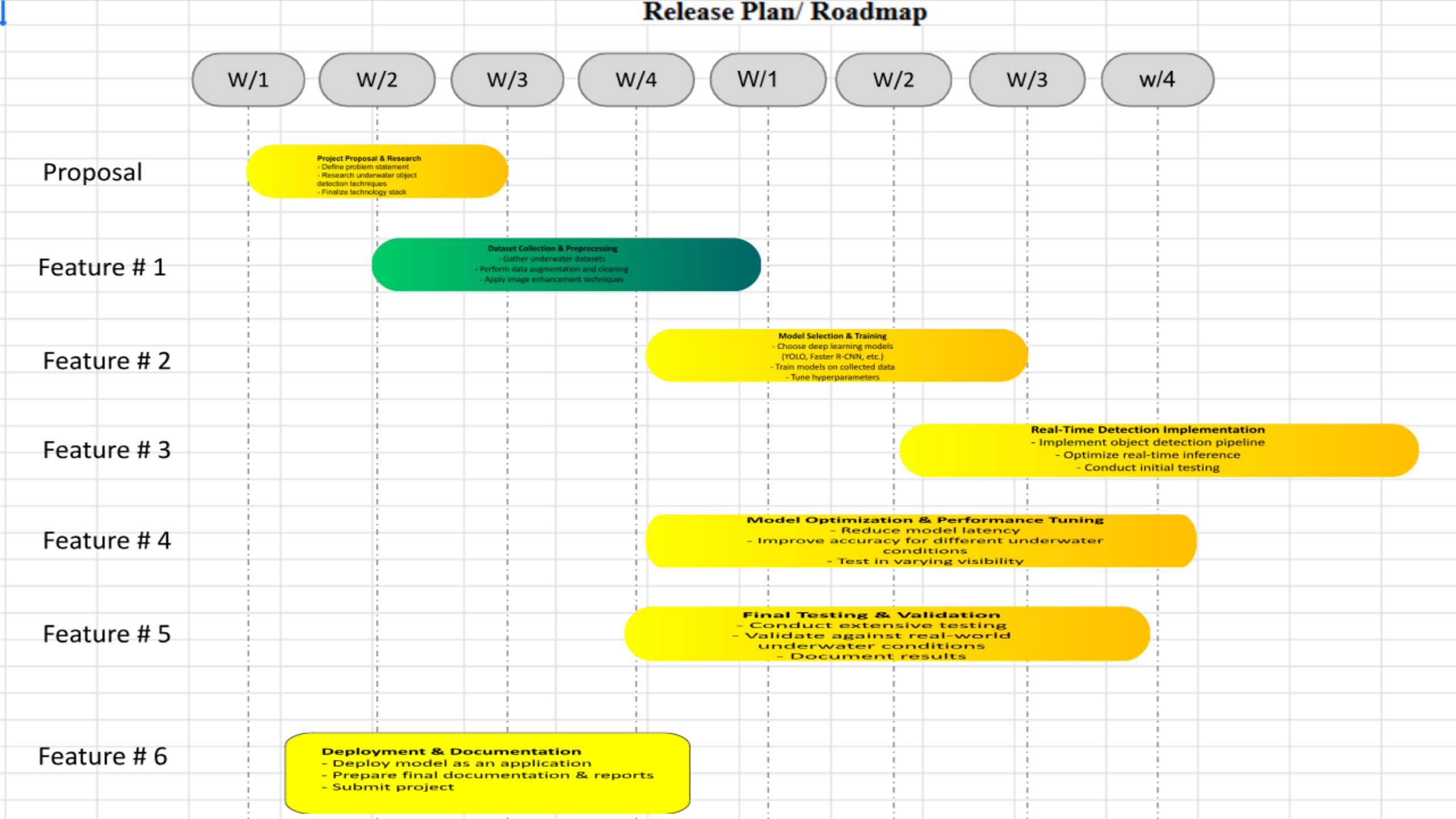
Product Backlog Refinement

Product Backlog for Under Water Object Detection Using YOLO

ID	Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	Dataset Collection	Data Collection	As a researcher, I want to collect underwater images/videos to train the model.	Must Have	In Progress	Dataset should contain diverse underwater objects with labeled data.	Integrate datasets from real-world sources & generate synthetic data.	High-quality, noise-reduced images.	10	TBD
2	Image Preprocessing	Preprocessing & Augmentation	As a developer, I want to enhance image quality to improve detection accuracy.	Must Have	To Do	Image clarity should improve using denoising and contrast adjustment.	Apply filters, noise reduction, and color correction.	Should work efficiently on real-time data.	7	TBD
3	Model Development	Model Training	As a data scientist, I want to train a deep learning model to detect underwater objects.	Must Have	To Do	Model should achieve at least 85% accuracy in detection.	Use CNN-based architectures (YOLO, Faster R-CNN, etc.).	Should be optimized for efficiency and real-time processing.	15	TBD
4	Real-Time Object Detection	Real-Time Detection	As an engineer, I want to deploy the model for real-time underwater detection.	Should Have	To Do	Model should process video feeds with <1s latency.	Implement TensorRT or OpenVINO for acceleration.	Should be lightweight and deployable on edge devices.	12	TBD
5	Testing & Evaluation	Validation & Testing	As a tester, I want to evaluate the model's performance on unseen underwater environments.	Must Have	To Do	Model should generalize well across different water conditions.	Perform cross-validation and test on multiple datasets.	Ensure robustness against low visibility and varying lighting.	8	TBD
6	Marine Species Classification	Object Recognition	As an environmentalist, I want the system to classify marine species accurately so that I can monitor biodiversity.	Should Have	To Do	System should classify species with at least 80% accuracy.	Implement species classification with deep learning.	Ensure robustness in varying underwater conditions.	10	TBD
7	Autonomous Navigation	Underwater Robotics	As a robotics engineer, I want the system to integrate with underwater drones for autonomous navigation.	Could Have	To Do	AI should assist in obstacle detection and path planning.	Develop object detection integration for AUVs.	Must run efficiently on embedded hardware.	14	TBD
8	Threat Detection	Security & Surveillance	As a naval security officer, I want an AI-driven detection system to identify potential underwater threats.	Could Have	To Do	System should detect unknown or suspicious objects with high recall.	Implement AI-based anomaly detection.	False positive rate should be <10%.	12	TBD
9	Depth & Position Analysis	Research & Oceanography	As an oceanographer, I need the system to analyze object depth and positioning for better research insights.	Should Have	To Do	System should estimate object depth with at least 90% accuracy.	Implement depth estimation using stereo vision.	Efficient processing for real-time applications.	9	TBD
10	Model Optimization	Performance Tuning	As an AI researcher, I want to evaluate and improve the system's deep learning model to enhance accuracy.	Must Have	To Do	Model accuracy should improve while keeping processing time low.	Optimize model using quantization and pruning.	Ensure performance remains stable across various conditions.	10	TBD

TIMELINE

Release Plan/ Roadmap



Implementation of Agile Board in MS Planner

My Plans > Agile Board | Grid | Board | Schedule | Charts | Timeline | ... | Share

Filters | Group by Bucket

Product Backlog (User Stories)

+ Add task

Due MV

Sprint 3 | User Story | Non Functional | Functional | Must have

Model Development

Model Training
As a data scientist, I want to train a deep learning model to detect underwater objects.
Acceptance Criteria:...

1 | 1 / 2

Due HA

User Story | Non Functional | Functional | Sprint 2 | Could have

Threat Detection

System should detect unknown or suspiciou

Sprint Backlog (In Progress)

+ Add task

Should be lightweight and deployable on ed

0 / 3

Due MV

User Story | Non Functional | Functional | Sprint 2 | Must have

Testing & Evaluation

Model should generalize well across differer
Perform cross-validation and test on multipl
Ensure robustness against low visibility and

0 / 3

Due TS

User Story | Sprint 4 | Non Functional | Functional | Should have

Marine Species Classification

Awaiting Review

+ Add task

User Story | Non Functional | Functional | ... | Sprint 2 | Must have

Dataset Collection

Data Collection
As a researcher, I want to collect underwater images/videos to train the model.
Acceptance Criteria:
Dataset should contain diverse underwater...

1 | 0 / 2

Due HA

Completed tasks 1

Completed Items

+ Add task

Sprint 3 | Sprint 4 | Sprint 2 | Sprint

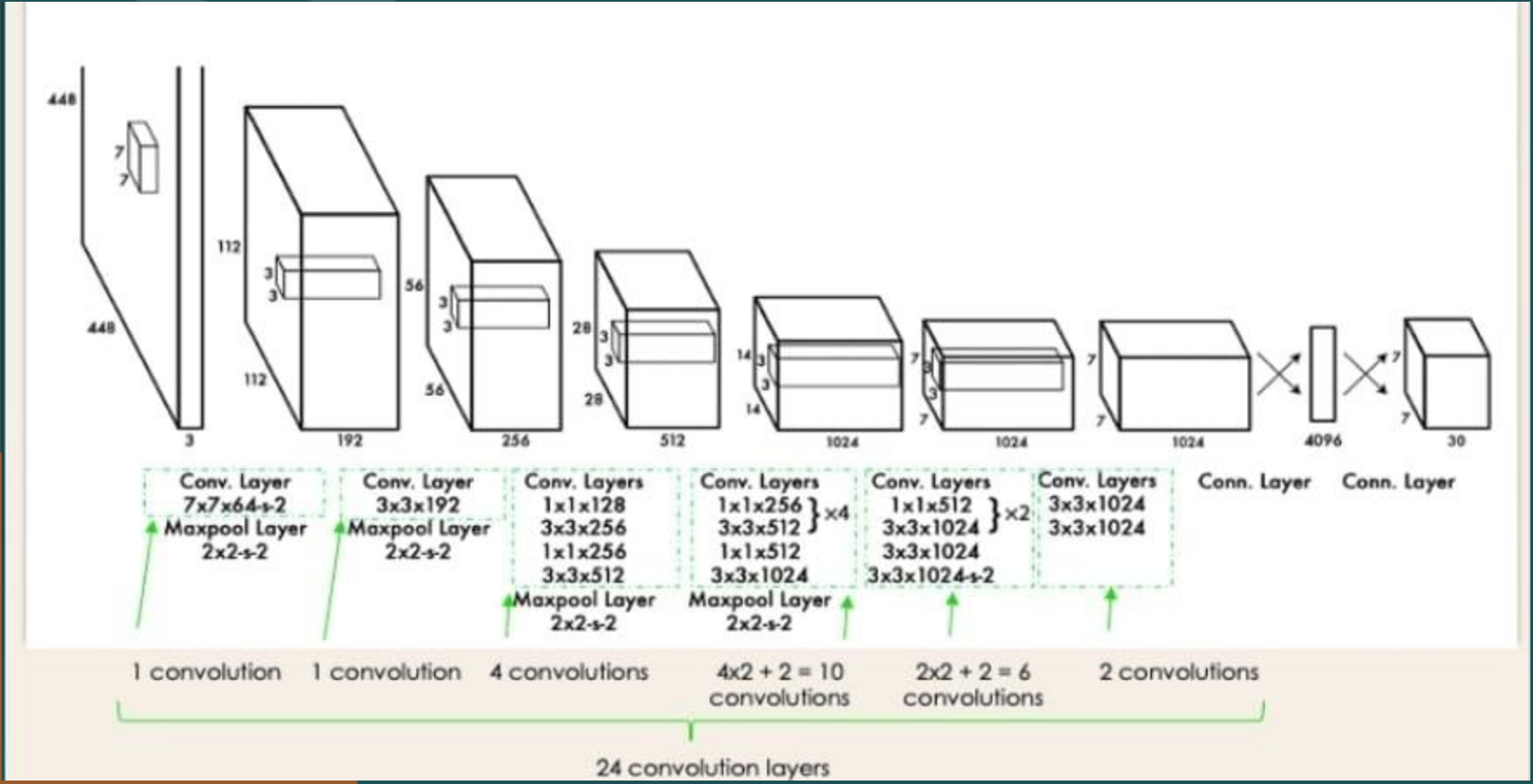
Release Plan

- Project Proposal & Research
- Dataset Collection & Preprocessing
- Model Selection & Training
- Real-Time Detection Implementation
- Model Optimization & Performance T
- Deployment & Documentation

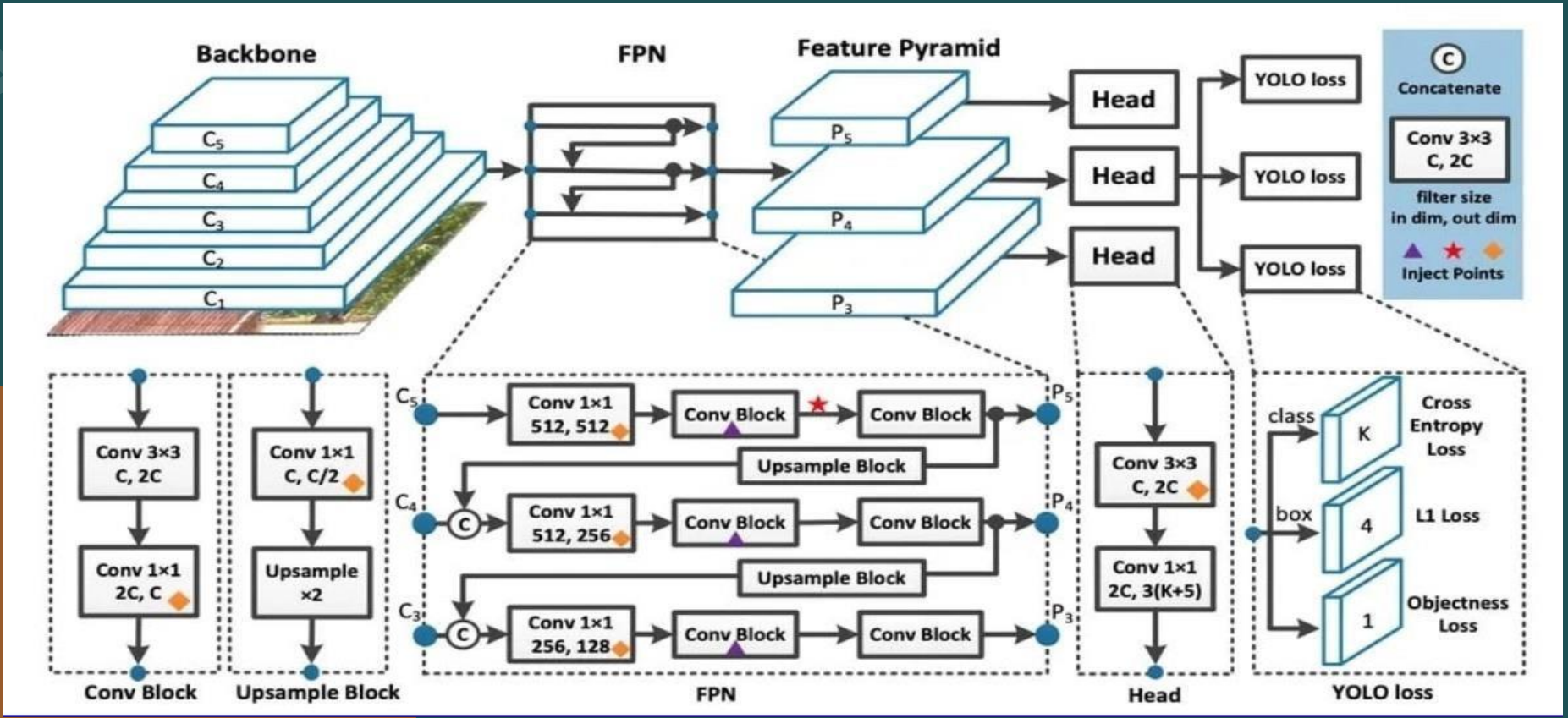
1 | 1 / 7

Due

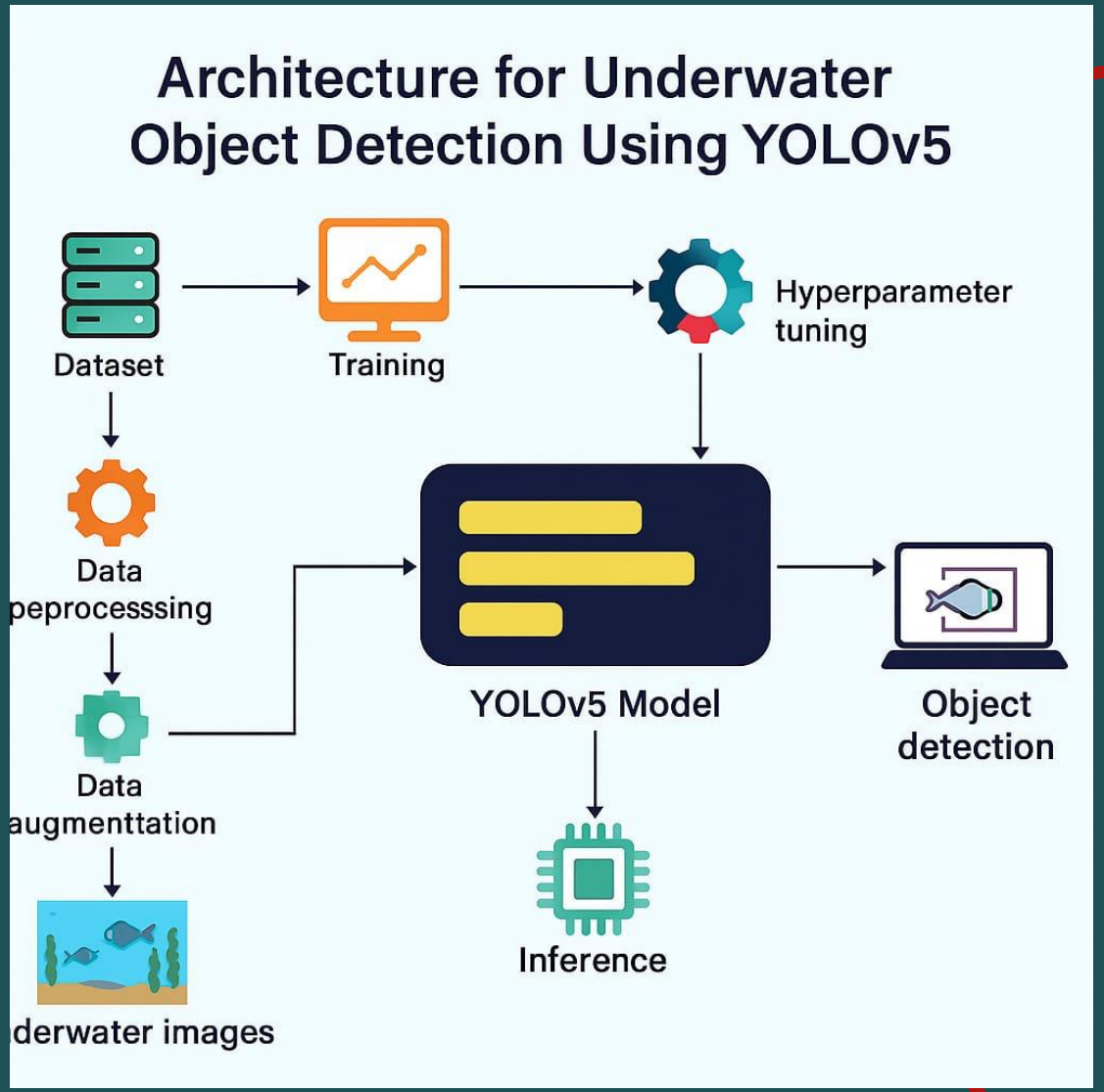
Yolo Architecture



Yolo Architecture



Architecture

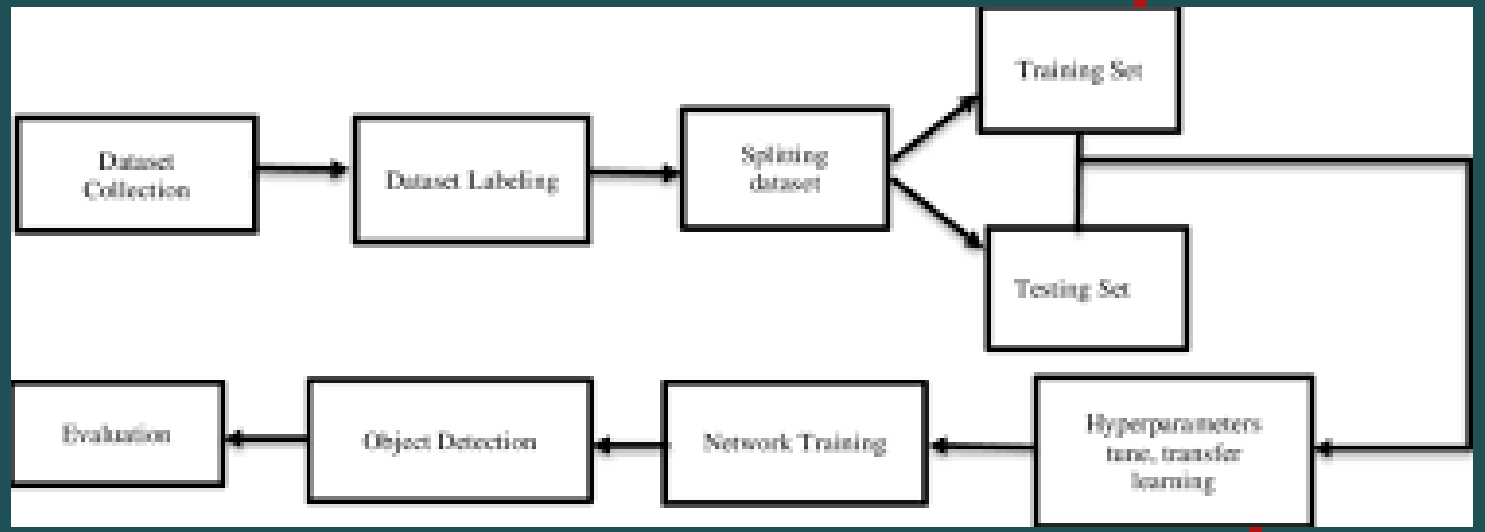


Data Flow Diagrams

BLOCK DIAGRAM

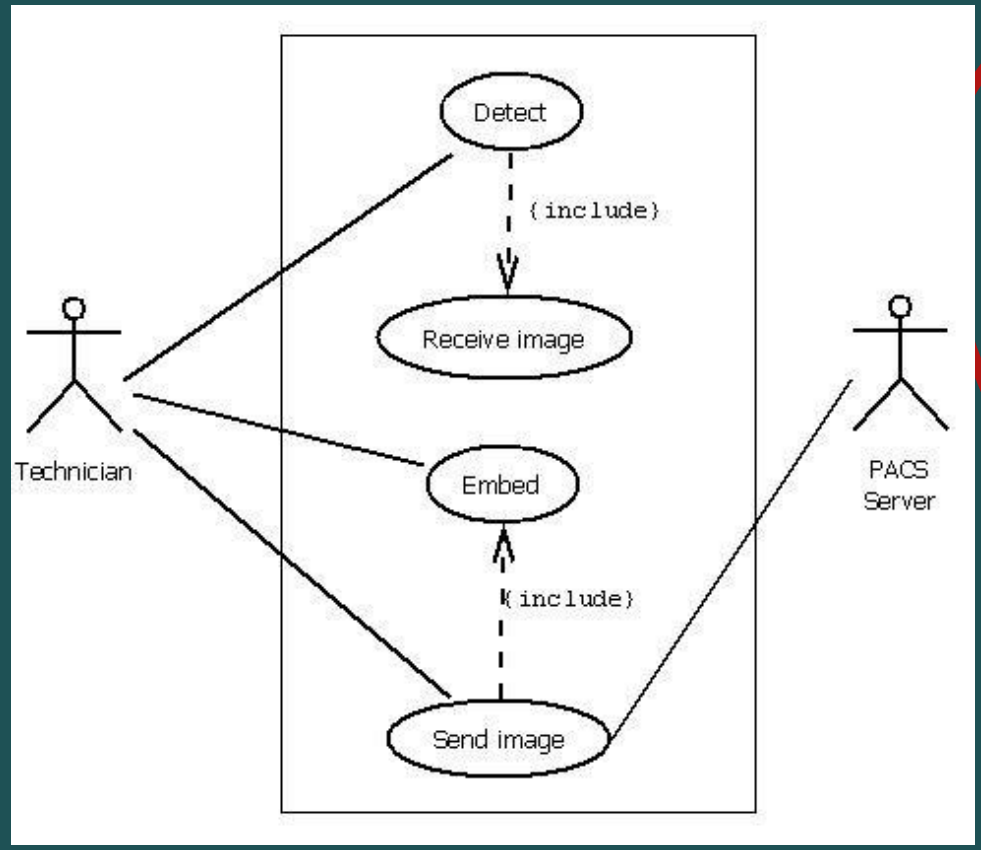
DFD LEVEL 0

DFD LEVEL 1

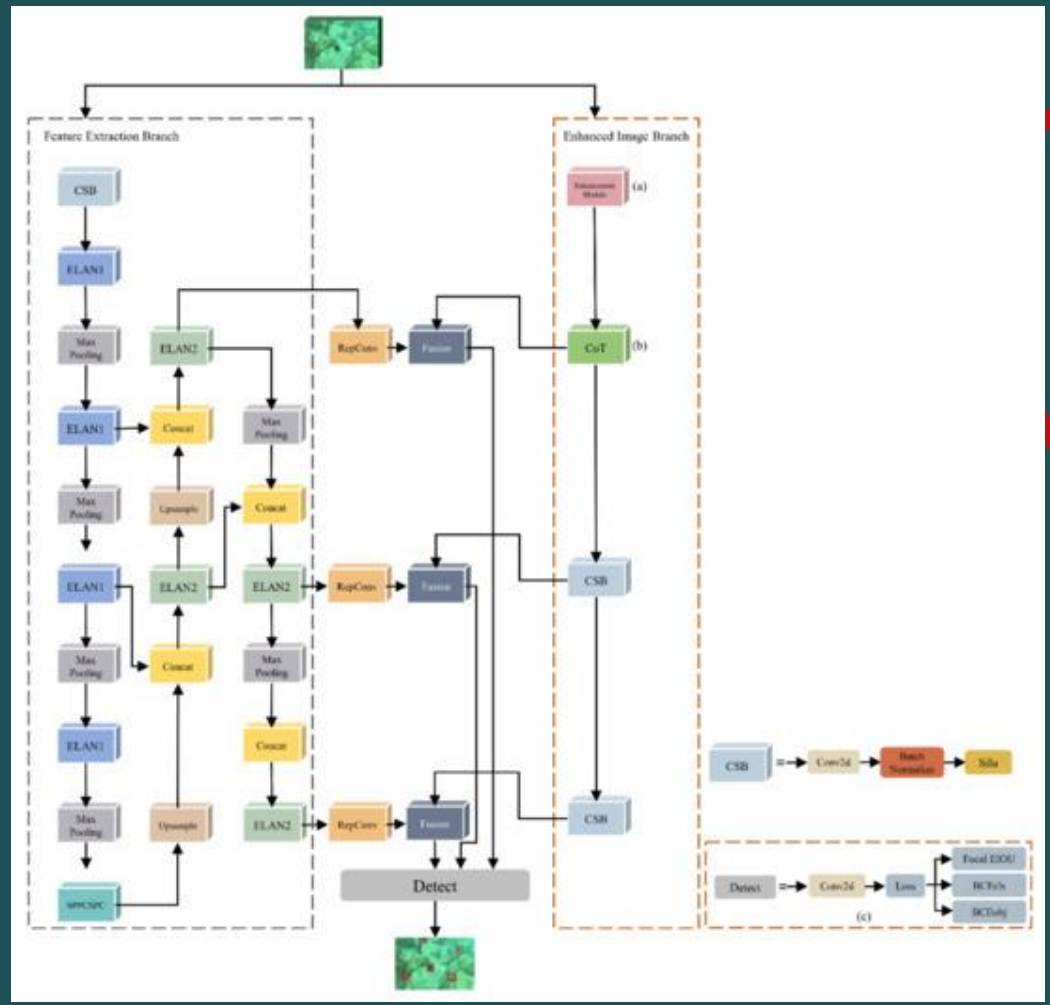

Dataset Collection
Dataset Labeling
Splitting dataset
Training Set
Testing Set
Hyperparameters tune, transfer learning
Network Training
Object Detection
Evaluation

UML DIAGRAMS

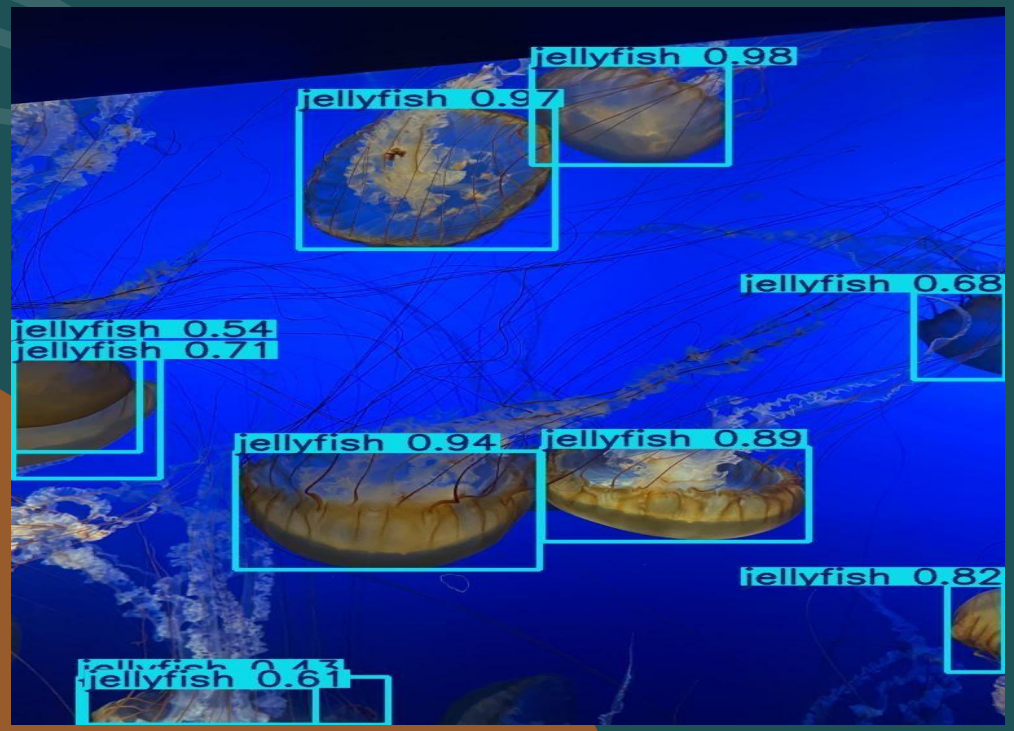
USE CASE

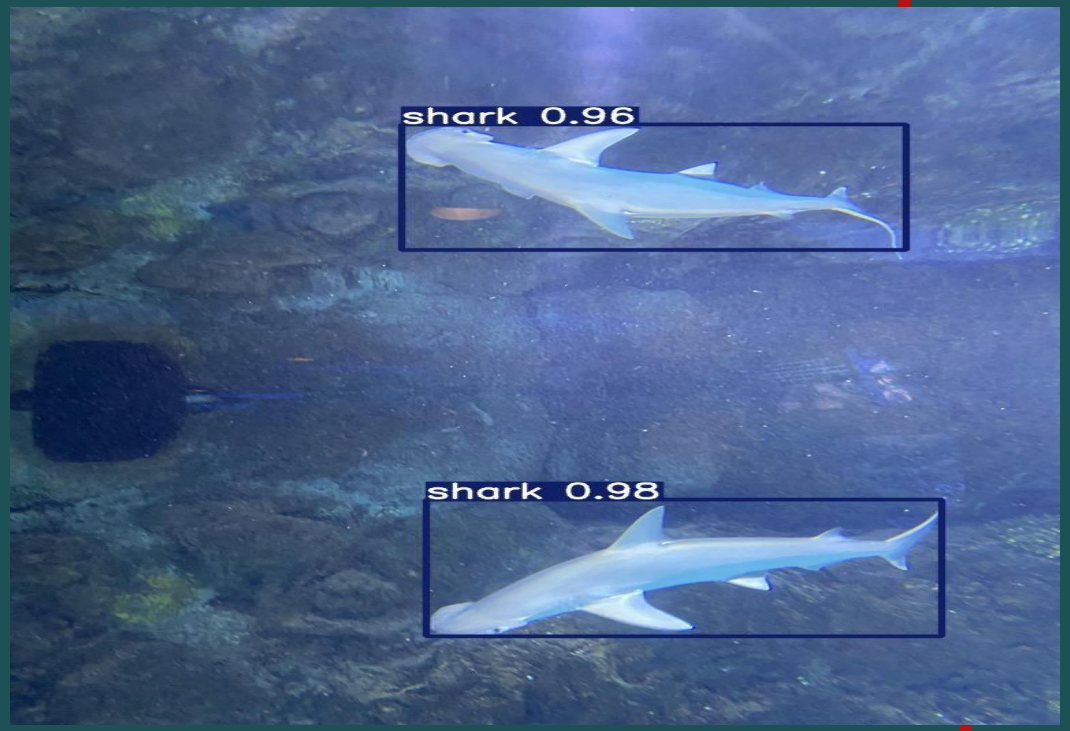

Detect
(include)
Receive image
Technician
Embed
(include)
Send image
PACS
Server

Sequence Diagram



RESULTS


jellyfish 0.98
jellyfish 0.97
jellyfish 0.68
jellyfish 0.54
jellyfish 0.71
jellyfish 0.94
jellyfish 0.89
jellyfish 0.82
jellyfish 0.61


shark 0.96
shark 0.98

THANK YOU