**Autonomous Drone-Based Real-Time Maritime Search and Rescue Using Onboard CV and Precision Payload Deployment**

**Abstract**

Maritime Search and Rescue (SAR) operations are inherently time-critical and demand rapid response over vast and often unpredictable oceanic environments. Traditional manual and semi-automated methods frequently fall short in delivering the speed, scalability, and accuracy required for effective rescues. This research presents an autonomous drone-based system equipped with an advanced computer vision model for enhancing SAR effectiveness through timely detection and intervention. The proposed system responds to emergency alarms, autonomously navigates to the incident location, hovers at a stable altitude of 200 meters, identifies individuals who have fallen into the water using an onboard vision model, and precisely deploys a life buoy to aid in rescue. To validate the detection component, we evaluate the performance of state-of-the-art object detection models, specifically the YOLOv8n and YOLOv11n models, trained on the SeaDronesSee dataset. Among the tested models, YOLOv8n and YOLOv11n demonstrated high accuracy with mAP50 scores of 0.680 and 0.688 respectively. Notably, YOLOv8n achieved better overall performance in terms of F1 score (0.691), accuracy (0.576), and computational efficiency (164.8 GFLOPs), making it more suitable for real-time deployment on edge devices onboard drones. The study highlights the promising synergy between autonomous drone platforms and advanced machine vision, offering a scalable and resource-efficient solution for real-time victim detection and life-saving interventions in maritime environments.

**Keywords:** Maritime Search, Autonomous, Drones, YOLOv8, YOLOv11, Machine Vision, Rescue Operations, Precision Payload Deployment

**1. Introduction**

The open sea, vast, unpredictable, and often unforgiving, has long presented significant challenges to human safety. From shipwrecks to individuals accidentally falling overboard, maritime environments frequently become the setting for life-threatening emergencies. In such scenarios, Maritime Search and Rescue (SAR) operations play a critical role in saving lives. However, these operations are often hindered by the sheer scale of the ocean, unpredictable weather conditions, and the difficulty in locating victims promptly. Traditional SAR methods, whether manual or semi-automated, struggle to meet the urgency and precision demanded by these life-critical situations, underscoring the need for more advanced, responsive, and efficient systems. Recent technological advancements offer promising alternatives.

Among these, Unmanned Aerial Vehicles (UAVs), commonly known as drones, stand out as transformative tools. Drones are flying machines that operate either autonomously or under remote control without onboard human presence. Based on their structural and functional characteristics, UAVs can be classified by size (mini, small, large) and are employed in diverse applications ranging from aerial photography and agriculture to infrastructure monitoring and emergency response. While large drones are often reserved for military applications, the increased accessibility and versatility of mini and small UAVs (weighing up to 25 kg) have accelerated their adoption in commercial and research domains.

In particular, autonomous drones, those capable of operating without constant human input, leverage onboard systems such as GPS, inertial measurement units (IMUs), and real-time control loops to navigate and execute complex tasks. These capabilities make them ideal for time-sensitive, high-risk applications like maritime SAR. Given the difficulty of accessing and monitoring vast oceanic regions, integrating drone technology can significantly improve the speed, precision, and effectiveness of victim detection and rescue response.

Furthermore, the integration of computer vision (CV) and artificial intelligence (AI) has elevated the capabilities of drones in autonomous operations. Object detection algorithms, especially real-time models like those in the YOLO (You Only Look Once) family, enable drones to identify and track people or objects in challenging marine environments. These models can be optimized for edge deployment, making real-time onboard inference feasible.

To evaluate these capabilities, we assess the performance of state-of-the-art object detection models, specifically the YOLOv8n and YOLOv11n models, trained on the SeaDronesSee dataset, a publicly available dataset designed for person-in-water detection from aerial views. Among these, YOLOv8n demonstrated better overall performance, achieving an F1 score of 0.691, accuracy of 0.576, and mAP50 of 0.680, compared to YOLOv11n’s F1 score of 0.732, accuracy of 0.569, and mAP50 of 0.688. Despite YOLOv11n showing slightly higher mAP50, YOLOv8n's stronger F1 score and accuracy, along with its efficient computational footprint (164.8 GFLOPs), make it more suited for real-time onboard deployment.

This research contributes to the growing field of autonomous drone-assisted SAR systems by developing a complete response pipeline: from the triggering of an alarm to drone deployment, real-time victim detection, and autonomous delivery of a life buoy to the victim’s location. By combining machine vision and UAV technology, this work highlights a scalable and practical approach to improving real-time maritime rescue effectiveness. The findings from this study aim to inform future deployment strategies and help overcome limitations of current SAR operations.

**1.1 Background Motivation**

Maritime regions like the Mediterranean Sea have long faced recurring tragedies involving overcrowded vessels and migrants in distress. In one instance, 136 lives were lost in September 2012, followed by 339 deaths in October 2023, and a later incident where 34 victims drowned despite 150 being rescued by the Maltese Navy. These events highlight the urgent need for faster, more efficient maritime Search and Rescue (SAR) systems. Closer to home, India’s 7,500 km coastline sees frequent maritime emergencies. Incidents such as the 2018 Cyclone Ockhi, where over 60 fishermen went missing, and the 2022 Mumbai cargo vessel capsize underline the limitations of current SAR methods. Coastal states like Goa, Kerala, and Tamil Nadu also report regular drowning cases due to strong currents and inadequate response time. Traditional SAR operations often suffer from slow deployment, limited visibility, and the vast scale of open waters. In critical rescue situations, delays cost lives. This research is driven by the need to enhance SAR capabilities through autonomous drone systems equipped with real-time object detection. By integrating YOLO-based machine vision models, specifically YOLOv8n and YOLOv11n, the proposed system aims to locate drowning victims and autonomously deliver life buoys. Among the two, YOLOv8n demonstrates superior overall performance, making it a more suitable choice for deployment in real-world maritime emergencies. The solution offers a scalable and rapid-response approach to improving the effectiveness of maritime rescue operations.

**1.2 Dataset**

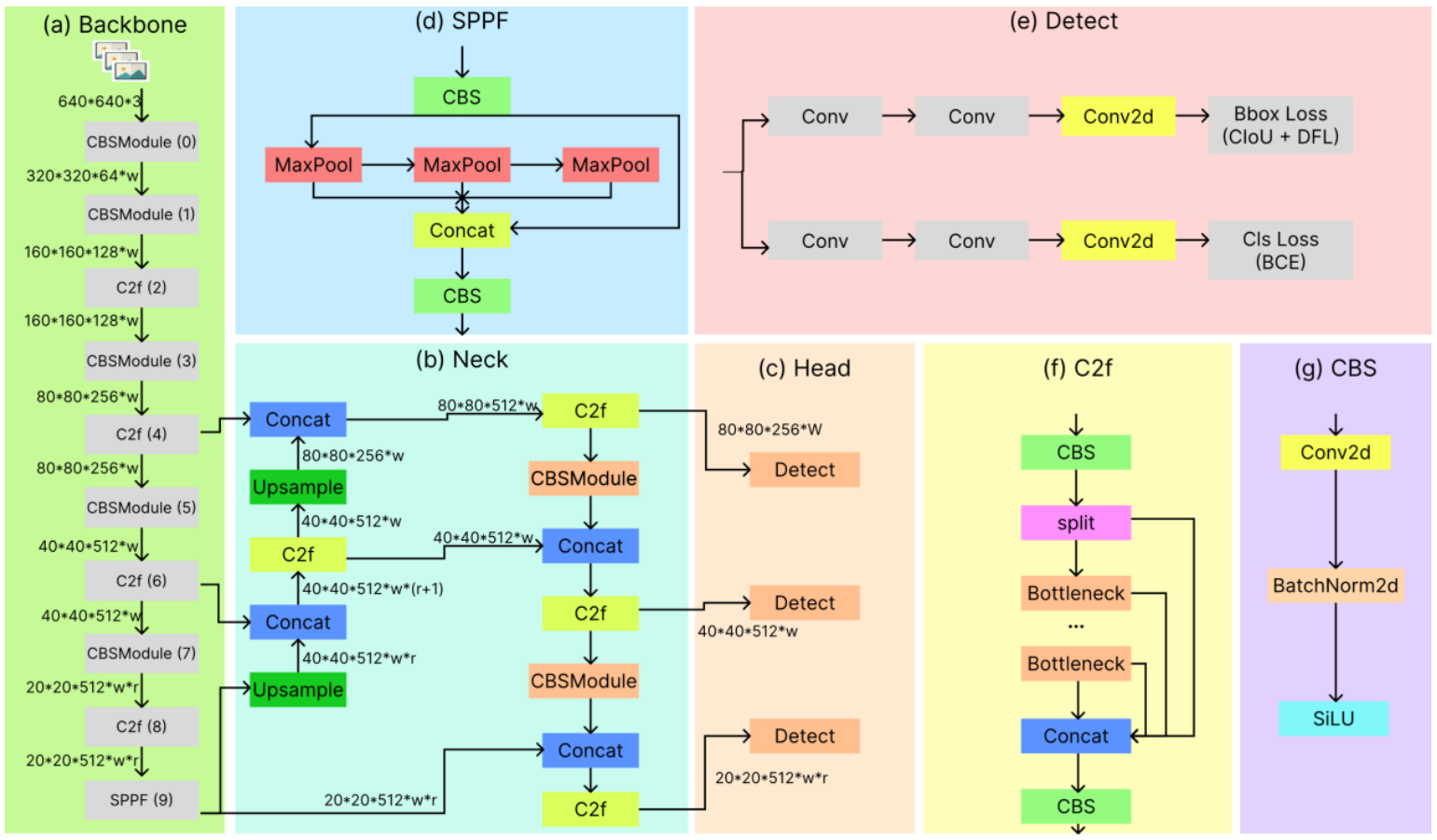
The dataset employed in this study is a modified version of the SeaDronesSee dataset, originally developed by researchers at the University of Tübingen. It comprises 10,474 images categorized into five classes: boat, buoy, jetski, life\_saving\_appliances, and swimmer. Captured by 20 different subjects, the dataset features aerial footage of open waters under varying lighting conditions, offering a realistic training ground for object detection in maritime environments. To enhance the robustness of the model, the dataset was further enriched using cutout augmentation, a data augmentation technique that improves generalization by randomly masking square regions within the input images. For training purposes, the dataset was divided using a 70-20-10 split for training, validation, and testing, respectively, resulting in 7,322 training images, 2,077 validation images, and 1,075 test images. These figures are detailed in Table 1.

**Table 1.** Dataset Partitioning

|  |  |  |
| --- | --- | --- |
| **Sub-set** | **Proportion** | **Images** |
| Train | 70% | 7,322 |
| Val | 20% | 2,077 |
| Test | 10% | 1,075 |

**1.3 YOLO Object Detection**

YOLO (You Only Look Once) is a widely adopted algorithm renowned for its high-performance object detection capabilities. Since the introduction of the YOLO series in 2015 [15], the framework has evolved through multiple versions, each offering notable improvements in speed and accuracy. In this study, the YOLOv8n and YOLOv11n models were trained and evaluated. The research involved extensive experimentation, including multiple rounds of parameter tuning and architectural adjustments, to identify an optimal model that delivers strong performance across evaluation metrics. Among the tested models, YOLOv8n demonstrated superior performance in terms of detection accuracy and overall balance between speed and resource efficiency.



**Figure 1.** Schematic figure of the network structure of YOLOV8

YOLOv8 employs a modified CSPDarknet53 architecture as its backbone, progressively downsampling features across five stages (B1 through B5) to generate multi-scale feature maps. A key innovation is the use of the lightweight C2f module, replacing the original Cross Stage Partial (CSP) module. The C2f module includes a gradient shunt connection that facilitates better information flow during feature extraction while maintaining computational efficiency.

Feature extraction in YOLOv8 is supported by the CBS module, which performs a sequence of convolution, batch normalization, and SiLU activation. At the end of the backbone, a Spatial Pyramid Pooling – Fast (SPPF) module condenses feature maps into fixed-size representations using three sequential max-pooling layers. This approach offers reduced latency compared to traditional SPP methods.

For feature aggregation, YOLOv8 uses a PAN-FPN neck, inspired by PANet. Unlike earlier versions such as YOLOv5 and YOLOv7, YOLOv8 omits convolution layers after upsampling, reducing architectural complexity without compromising performance. The dual-path PAN-FPN design combines features from both top-down and bottom-up flows, effectively merging deep semantic information with spatial detail to support accurate object localization.

The detection head in YOLOv8 uses a decoupled architecture with separate branches for classification and bounding box regression. Binary Cross Entropy (BCE) loss is used for classification, while a combination of Distribution Focal Loss (DFL) and Complete IoU (CIoU) loss is used for bounding box refinement. YOLOv8 also adopts an anchor-free approach and employs a Task-Aligned Assigner to dynamically determine positive and negative samples during training, thereby enhancing both accuracy and robustness.

Building upon the architecture and design principles of YOLOv8, YOLOv11 introduces further enhancements for detection and segmentation tasks. It features an optimized backbone and neck structure that improves feature extraction while maintaining efficiency and adaptability. While YOLOv11n also delivers competitive results, YOLOv8n outperformed it overall in this study. YOLOv11 maintains a lightweight design and compatibility with edge and cloud platforms, making it suitable for a broad range of deployment scenarios. It supports advanced tasks including oriented object detection and segmentation, offering flexibility for high-performance, real-time applications.

**2. Literature Survey**

**2.1 Object Detection in Rescue Operations**

Recent advancements have seen the integration of drone-based object detection systems in various rescue scenarios. For instance, YOLOv4 has been effectively used to detect individuals involved in accidents during high-risk outdoor activities such as skiing, hiking, and mountain biking. To overcome the limitations of traditional vision-based systems in low-visibility conditions, Thermal Infrared (TIR) cameras have been employed for automatic human detection in search and rescue (SAR) missions. Additionally, convolutional neural network (CNN) models have been developed to detect critical ground features from aerial imagery in post-disaster environments. These models, trained on the custom Volan2018 aerial video dataset, are capable of identifying damaged and intact rooftops, vehicles, vegetation, debris, and flood zones, demonstrating the effectiveness of CNNs in disaster response and assessment.

**2.2 Summary of SeaDronesSee dataset in Rescue tasks**

The SeaDronesSee dataset was created to address the lack of suitable datasets tailored for maritime search and rescue (SAR) operations. Previous datasets focused primarily on remote sensing using synthetic aperture radar (SAR) imagery, which relied on satellite-captured top-down views. While effective for detecting large vessels, these datasets are inadequate for identifying smaller objects such as swimmers. Additionally, satellite imagery is often hindered by environmental factors like cloud cover, reducing its reliability in time-critical rescue missions. In contrast, the SeaDronesSee dataset offers high-resolution RGB imagery, ranging from 3840 × 2160 px to 5456 × 3632 px, and includes dedicated object classes such as boats, jet skis, buoys, life-saving appliances, and swimmers, making it more suitable for close-range drone-based SAR tasks.

**2.3 Object Detection models utilizing the SeaDronesSee dataset**

The SeaDronesSee dataset was created to address the lack of suitable datasets tailored for maritime search and rescue (SAR) operations. Previous datasets focused primarily on remote sensing using synthetic aperture radar (SAR) imagery, which relied on satellite-captured top-down views. While effective for detecting large vessels, these datasets are inadequate for identifying smaller objects such as swimmers. Additionally, satellite imagery is often hindered by environmental factors like cloud cover, reducing its reliability in time-critical rescue missions. In contrast, the SeaDronesSee dataset offers high-resolution RGB imagery, ranging from 3840 × 2160 px to 5456 × 3632 px, and includes dedicated object classes such as boats, jet skis, buoys, life-saving appliances, and swimmers, making it more suitable for close-range drone-based SAR tasks.

**3. Methodology**

**3.1 Dataset Preparation**

A refined version of the SeaDronesSee dataset, originally comprising 5,630 annotated images, was selected for this study. These images were captured using five distinct camera systems (see Table 2) mounted on three different drones, DJI Matrice 100, DJI Matrice 210, DJI Mavic 2 Pro, and a fixed-wing Trinity F90+ aircraft developed by Quantum Systems [13]. The diversity of imaging platforms was intended to minimize camera bias and ensure broader generalization across different visual conditions.  
The initial annotations were created using DarkLabel, a free and open-source labeling tool, and classified into five categories: swimmer (person in water without a life jacket), floater (person in water with a life jacket), swimmer† (person on boat without a life jacket), floater† (person on boat with a life jacket), and boats.

**Table 2.** Specifications of cameras used in generating the SeaDronesSee dataset

|  |  |  |
| --- | --- | --- |
| **Camera** | **Resolution** | **Purpose** |
| Hasselblad L1D-20c | 3840 \* 2160 | Video capture at 30 fps |
| MicaSense RedEdge-MX | 1280 \* 960 | Multi-spectral capture at 1 fps |
| Sony UMC-R10C | 5456 \* 3632 | Image capture |
| Zenmuse X5 | 3840 \* 2160 | Video capture at 30 fps |
| Zenmuse XT2 | 3840 \* 2160 | Video capture at 30 fps |

For this research, we adopted the Roboflow SeaDronesSee v10 dataset, an augmented and reorganized version of the original. This version features 10,474 images, expanded through cutout augmentation and auto-orientation techniques. The annotation schema was restructured to define five new consolidated classes: boat, buoy, jetski, life\_saving\_appliances, and swimmer. Notably, the ‘swimmer’ class aggregates all four person-related categories from the original dataset. The class-wise distribution of these merged categories is shown in Table 3, and their instance frequencies in the training dataset are visualized in Figure 2.

**Table 3.** Frequency of class image across dataset images

|  |  |
| --- | --- |
| **Class** | **Images** |
| Swimmer | 8,185 |
| Boat | 6,782 |
| Buoy | 4,073 |
| Jetski | 2,648 |
| Life\_saving\_appliances | 856 |

**3.2 Conceptual Framework for Autonomous Aerial Life Buoy Deployment**

This research proposes a conceptual autonomous drone-based rescue system specifically tailored for maritime search and rescue (SAR) missions involving individuals who have fallen overboard or into nearshore waters. The envisioned pipeline integrates emergency detection, autonomous navigation, real-time object detection, and targeted payload deployment, forming a closed-loop rescue response system. While not physically implemented in this phase of the study, each component of the system is designed to be grounded in practical deploy-ability and informed by current state-of-the-art computer vision and robotics frameworks.

**i. Incident Trigger and Alarm Interface**

The operational cycle begins with the triggering of an emergency alarm. This alert may originate from a variety of sources, including wearable sensors on passengers, ship-based fall-detection systems, coastal surveillance units, or visual verification by human observers. Upon confirmation, the alert is relayed to a central command module interfaced with a fleet of standby unmanned aerial vehicles (UAVs).

**ii. Autonomous Drone Dispatch and Path Planning**

Upon receiving the incident coordinates, a rescue UAV is immediately deployed. The drone is programmed to execute an autonomous take-off routine, followed by real-time path planning toward the alarm zone. The drone's navigation system leverages GPS and pre-mapped geospatial data to compute an optimal route, balancing speed and safety. The route planning logic incorporates several modules:

* **Waypoint Generation**: A set of dynamic waypoints are generated between the drone's base station and the target location, considering both geodetic distance and environmental constraints (e.g., no-fly zones, restricted airspace, or weather conditions).
* **Obstacle Avoidance**: Using onboard LiDAR, radar, or depth-sensing cameras (in future hardware integrations), the UAV can dynamically adjust its path in response to sudden obstructions such as terrain features, tall structures, or other aircraft.
* **Adaptive Altitude Control**: The UAV maintains an altitude of approximately 200–250 meters over water to ensure a wide field of view for visual detection while preserving spatial resolution for effective object recognition by the onboard camera.

The drone continues to the specified alarm region, hovering and loitering in a circular or grid-based search pattern if no swimmer is immediately detected upon arrival.

**iii. Visual Search and Detection Using Onboard Computer Vision**

Once over the search area, the UAV activates its downward-facing RGB camera system, calibrated to capture high-resolution imagery compatible with the inference requirements of the detection model. The onboard YOLOv8n object detection model, pretrained and fine-tuned on the SeaDronesSee v8 dataset, is deployed to scan each video frame in real time.

Key features of the detection phase include:

* **Frame-wise Inference:** The RGB video stream is processed frame-by-frame at 30 FPS, with bounding boxes, class labels, and confidence scores logged per frame.
* **Detection Validation:** To mitigate false positives due to water reflections, waves, or floating debris, the system uses temporal smoothing logic. A detection is confirmed only if the same object class (i.e., swimmer) appears with a confidence ≥ 0.7 in at least three consecutive frames.
* **Localization and Target Tracking:** Upon confirmation, the centroid coordinates of the bounding box are extracted and converted into geospatial coordinates using camera intrinsics and GPS-altitude metadata. The drone then switches to a target-following mode, continuously updating the swimmer's location as they drift or move with the waves.

**Figure 2.** Drone Navigating towards Stranded People over the Ocean

**iv. Precision Payload Deployment**

Once the swimmer’s location is stabilized within a defined spatial tolerance (e.g., deviation < 2 meters across three frames), the UAV repositions directly over the centroid of the target. The drone transitions to hover mode, maintains positional lock using GPS and barometer fusion, and prepares for payload deployment. The payload mechanism is designed to release a single-use life-saving buoy.

Though this research phase does not implement the mechanical deployment, the complete logic for trajectory prediction and visual confirmation is established, forming a reliable basis for future integration with drop-capable UAV platforms.

**v. Operational Evaluation Metrics**

The performance of the system is intended to be evaluated based on three primary metrics:

* **Detection Accuracy:** Derived from the YOLOv11n model’s swimmer-class precision, recall, and mAP metrics as benchmarked on SeaDronesSee v10.
* **Drop Precision:** Defined as the Euclidean distance between the predicted swimmer coordinates and the actual point of buoy impact in water.
* **System Latency:** Measured from the timestamp of swimmer entry in the visible field to the moment of detection confirmation and drop execution.

**Figure 3.** Drone Carrying Life Jacket Payload

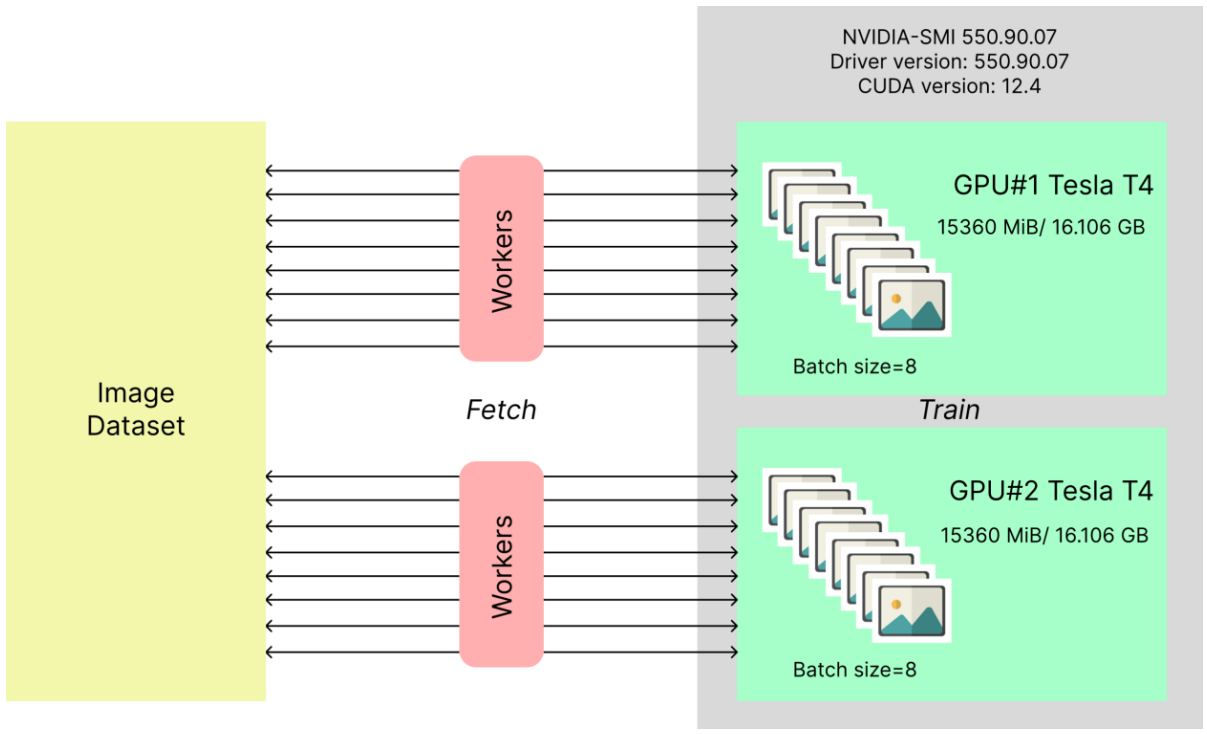
This autonomous rescue pipeline emphasizes speed, accuracy, and real-time responsiveness, and is structured to operate effectively in dynamic, high-risk maritime environments. The methodology, while conceptual in this paper, offers a viable roadmap for future development and deployment of aerial life-saving systems powered by deep learning and autonomous robotics.

**3.3 Model Configurations**

The configurations implemented in this study were carefully designed to thoroughly evaluate the performance of the YOLOv8 and YOLOv11 models under a wide range of experimental conditions. Each version was tested using various input image sizes, anchor settings, and hyperparameter combinations to identify the most effective setup for maximizing detection accuracy while minimizing computational cost. Particular attention was given to tuning critical hyperparameters such as learning rate, batch size, and number of training epochs to ensure the models achieved both precision and efficiency. For YOLOv8, the model was optimized to maintain a balance between speed and accuracy, while YOLOv11 incorporated a set of architectural advancements, including refined backbone structures and improved feature extraction mechanisms, to further boost detection capability and robustness across diverse scenarios.

The models that were ultimately selected for training and evaluation in this work are YOLOv8n.pt and YOLOv11n.pt. Given that search and rescue (SAR) operations often involve identifying small, dispersed, and low-resolution targets, such as swimmers or life-saving appliances, additional network adaptations were necessary to support better localization. Specifically, a high-resolution P2 layer was incorporated into the detection pipeline. This layer, added via the .yaml configuration file, enabled the network to detect small objects more effectively by increasing the spatial granularity in earlier stages of the model. The .yaml file also defined the train, validation, and test source directories for consistency in data handling.

Model training was conducted using the Kaggle platform, which provides access to advanced GPU infrastructure. For this study, Kaggle's compute environment offered limited-time usage of NVIDIA-SMI version 550.90.07, featuring a dual-GPU setup with Tesla T4 GPUs. As illustrated in Figure 3, the environment supports efficient multi-threaded data loading using parallel workers. These workers fetch and preprocess image batches, minimizing I/O delays and ensuring steady GPU utilization. The dataset was then distributed in parallel to the two GPUs, which enabled faster model training and reduced time-to-convergence due to distributed computation



**Figure 3.** NVIDIA-SMI 550.90.07

Model training was conducted using the Kaggle platform, which provides access to advanced GPU infrastructure. For this study, Kaggle's compute environment offered limited-time usage of NVIDIA-SMI version 550.90.07, featuring a dual-GPU setup with Tesla T4 GPUs. As illustrated in Figure 3, the environment supports efficient multi-threaded data loading using parallel workers. These workers fetch and preprocess image batches, minimizing I/O delays and ensuring steady GPU utilization. The dataset was then distributed in parallel to the two GPUs, which enabled faster model training and reduced time-to-convergence due to distributed computation.

All models were trained for a total of 100 epochs, which was found to be an optimal point for balancing model performance while mitigating risks of overfitting. Both the batch size and the number of workers were set to 8, aligning with the memory constraints of the GPU environment while also speeding up the data processing pipeline. To improve detection in crowded maritime scenes, an Intersection over Union (IoU) threshold of 0.7 was employed. This high threshold allowed for more accurate object distinction by reducing the overlap in predicted bounding boxes. The training leveraged the AdamW optimizer with the momentum setting kept to ‘Auto’, which dynamically adjusted during training. The final models, YOLOv8n and YOLOv11n, used a momentum value of 0.9 and a learning rate of 0.000714, selected based on iterative experimentation for stable and effective convergence across both model variants.

**3.4 Evaluation Metrics**

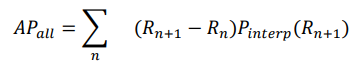
Across various annotated datasets employed by object detection challenges and the research community, the primary metric for evaluating detection accuracy is Average Precision (AP). To fully understand the variations of AP, it is essential to first familiarize ourselves with some fundamental terms commonly used in this context:

* True Positive (TP): A correct identification of a ground-truth bounding box.
* False Positive (FP): An incorrect identification, either detecting a non-existent object or misplacing the detection of an existing one.
* False Negative (FN): A failure to detect an actual ground-truth bounding box.

The count of True Negatives (TN) is irrelevant in object detection since the number of potential bounding that should not be identified within an image is infinite.

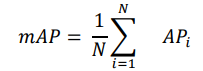
Precision and recall are important measures in machine learning that evaluate the performance of a model. Precision computes the correctness of positive predictions, representing the percentage of correct positive predictions. While recall determines how well the model recognizes all relevant instances.

A precision-recall curve (PR curve) is used for visualizing the relationship between Precision and Recall, across varying confidence thresholds assigned to the bounding boxes predicted by the detector.

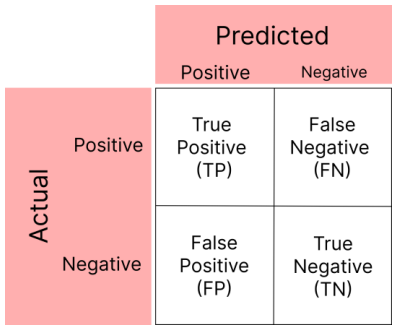
In the 11-point interpolation method, the precision-recall curve is summarized by averaging the highest precision values at 11 evenly spaced recall levels: 0, 0.1, 0.2, ..., 1. The AP is calculated by considering the maximum interpolated precision, Pinterp(R) at each recall level, rather than the observed precision, P(R).

where,

And 𝑅̃ denotes the mean of R values.

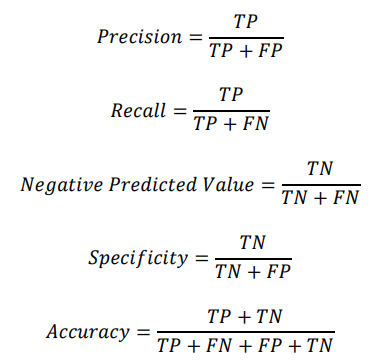
To measure the accuracy over all classes, the average of AP over all classes is taken, this metric is known as the mean average precision (mAP.)

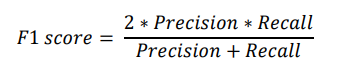
The average precision (AP) is a per-class measure while mAP is the average of APs across all the classes. Thus, mAP is a robust evaluation metric that considers multiple queries.

A confusion matrix is a tabular representation of a model’s performance. It compares the predicted labels to the actual labels and provides detailed insights into the model's performance in each class. For object detection, the confusion matrix is typically used for evaluating the classification of detected objects. Figure: 7 shows the structure of a confusion matrix.

**Figure 7.** Structure of a confusion matrix

The structure of a confusion matrix has been depicted in Figure 7, which serves as an important evaluation tool in classification tasks. It consists of four key components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The matrix helps analyse model performance, providing insights into accuracy, precision, recall, and overall error rates for the trained model.

Formulae of evaluation metrics like Precision, Recall, Negative Predictive Value, Specificity, and Accuracy, are as follows:

The F1 score is a metric used to measure the performance of a machine learning model. It can be calculated by combining the precision and recall of the model. The F1 score can be calculated in the following way:

**4. Result and Discussion**

**2.1 Object Detection in Rescue Operations**