**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 simulate maritime scenarios using Unreal Engine 4.2 and evaluate the performance of state-of-the-art object detection models, specifically the YOLOv8 and YOLOv11 series, trained on the SeaDronesSee dataset. Among the tested models, YOLOv11m, YOLOv8l, and YOLOv8x demonstrated the highest accuracy with mAP50 scores of 0.692, 0.687, and 0.680 respectively. Notably, YOLOv11m achieved superior performance while maintaining lower computational complexity (GFLOPs), making it 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, Unreal Engine, 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 in a controlled yet realistic setting, simulations offer a cost-effective and scalable solution. Computer simulations allow researchers to model rescue scenarios, validate machine learning models, and refine mission parameters without the risks or expenses of real-world trials. This study utilizes Unreal Engine 4.2, a powerful simulation platform originally developed for game development, to construct a realistic maritime SAR scenario. The simulated environment enables us to test and compare the performance of state-of-the-art object detection models, specifically YOLOv8 and YOLOv11, trained on the SeaDronesSee dataset, a publicly available dataset designed for person-in-water detection from aerial views.

This research contributes to the growing field of autonomous drone-assisted SAR systems by simulating 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 simulation, machine vision, and UAV technology, this work highlights a scalable and practical approach to improving real-time maritime rescue effectiveness. The insights gained from the simulation can 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 simulating SAR scenarios and integrating YOLO-based machine vision models, the proposed system aims to locate drowning victims and autonomously deliver life buoys, offering a scalable and rapid-response solution to maritime emergencies.

**1.2 Dataset**

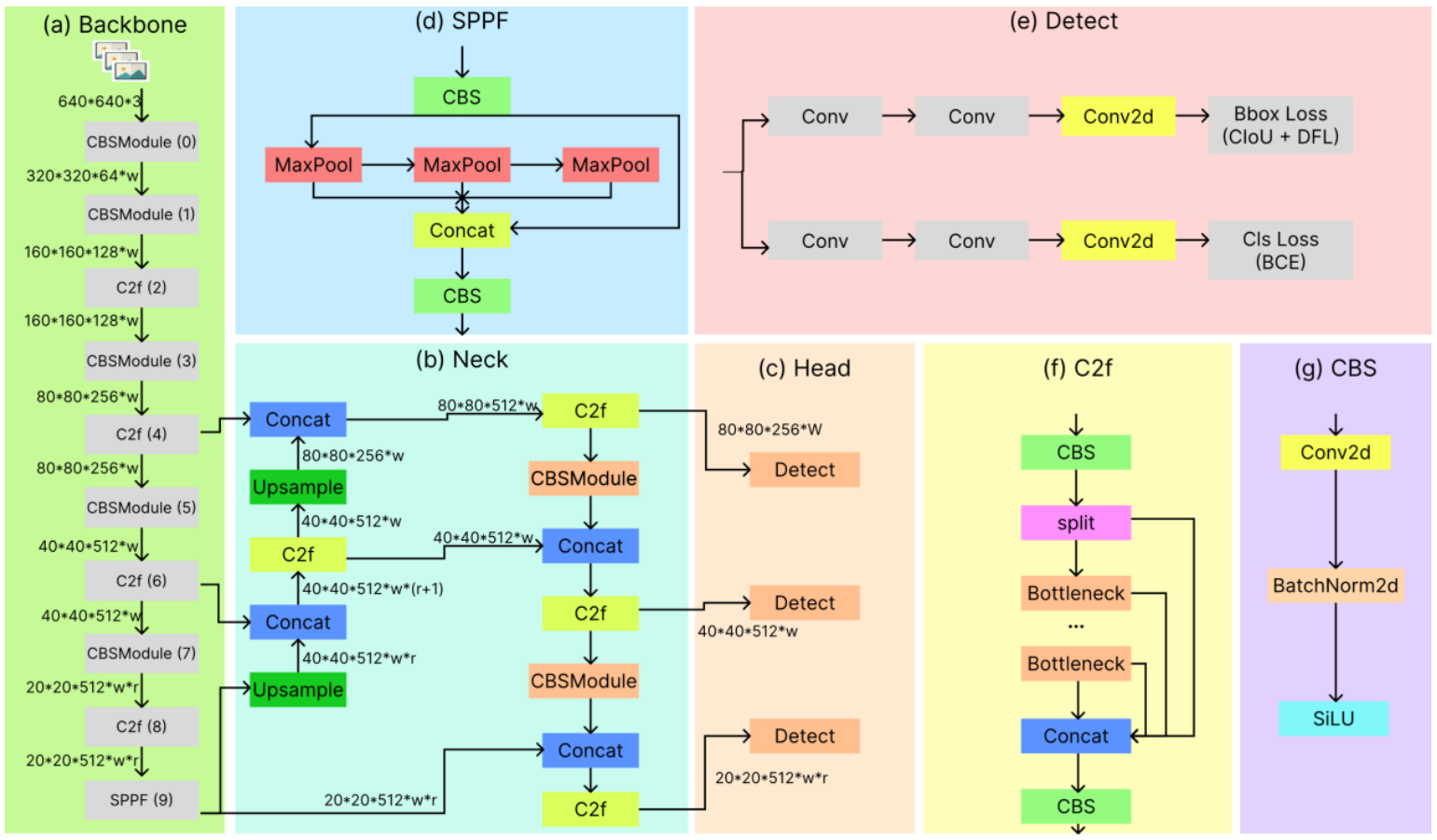
The dataset employed in this study is a modified version of the SeaDronesSee dataset, originally developed by researchers at the University of Tübingen [20]. It comprises 10,474 images categorized into five classes: boat, buoy, jetski, life\_saving\_appliances, and swimmer. Captured by 20 different subjects, 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, various model sizes from both the YOLOv8 and YOLOv11 series 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.



**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. YOLOv11 achieves higher mean Average Precision (mAP) scores on benchmark datasets such as COCO, despite having fewer parameters than models like YOLOv8m. Its lightweight design and compatibility with edge and cloud platforms make it suitable for a broad range of deployment scenarios. YOLOv11 supports advanced tasks including oriented object detection and segmentation, making it a powerful and flexible solution 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**

Several object detection models have been evaluated using the SeaDronesSee dataset, with performance varying significantly across architectures. Models based on Faster R-CNN, Cascade R-CNN, and RetinaNet have achieved AP50 scores ranging from 0.301 to 0.718, with the highest accuracy attained by the FR.ResNeXt-101-FPN-heuristic model. On the other hand, custom adaptations of the default YOLOv8 models have reported AP50 scores between 0.379 and 0.591. The relatively lower performance across models can largely be attributed to the dataset’s complexity, especially the presence of extremely small and often occluded targets, such as swimmers and life-saving appliances, which present a significant challenge for accurate detection during validation.

**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 [19], 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 |

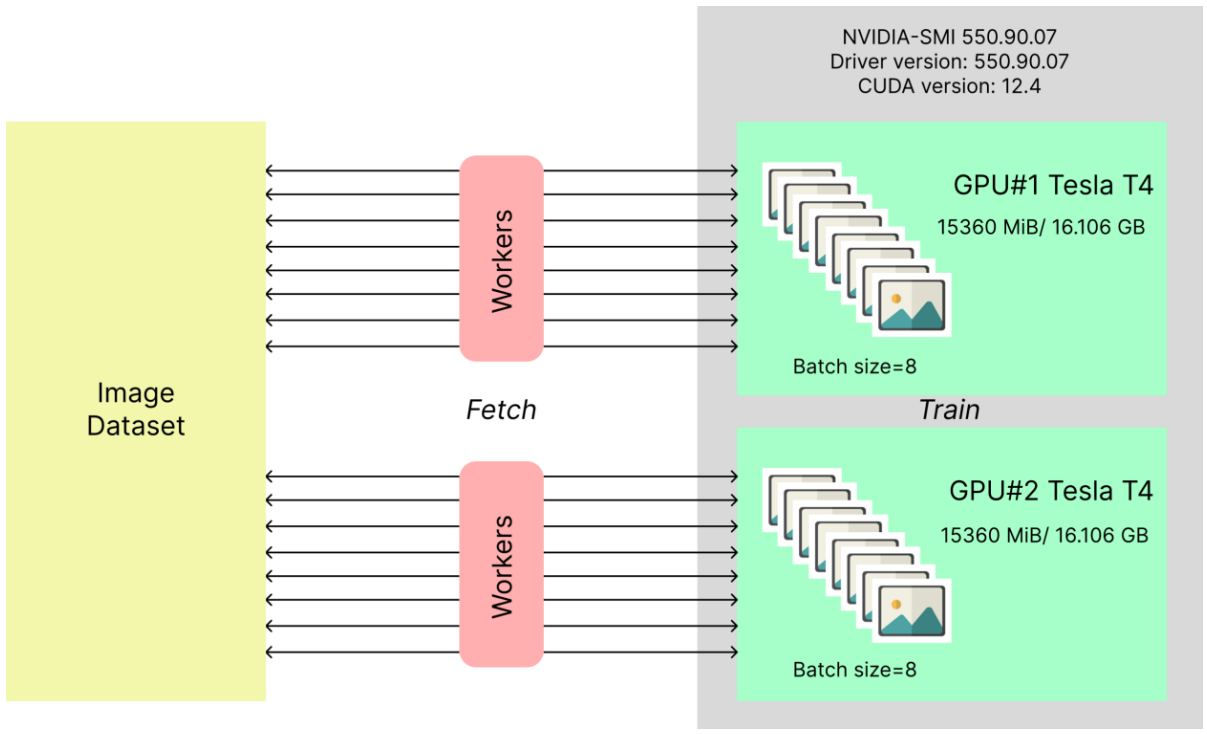
**3.3 Path Planning Implementation**

**3.4 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 YOLOv8m.pt and YOLO11m.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 50 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 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.