An AI-Driven Real-Time Vision-Based System for Underwater Object Detection and Tracking Using an ROV

Syed Shahareyar
Dept. of Aerospace Engineering & Autonomous Systems
Defence Institute of Advanced Technology
Pune, India
shahareyarsyed@gmail.com

Pooja Agrawal
Dept. of Aerospace Engineering & Autonomous Systems
Defence Institute of Advanced Technology
Pune, India
poojaa.diat@gmail.com

Sashi Kanta Panigrahi
Dept. of Mechanical Engineering
Defence Institute of Advanced Technology
Pune, India
panigrahi.sk@gmail.com

doi: https://doi.org/10.37285/bsp.SACAD2025.60

***Abstract*—This paper presents a unified, real-time AI based dashboard (UW-dash) for underwater inspection and monitoring using a Remotely Operated Vehicle (ROV). The proposed UW-dash processes underwater imagery captured from the ROV's onboard camera. The dashboard includes three integrated stages which are image enhancement, object detection, and multi-object tracking. Input image is enhanced using image processing Contrast Limited Adaptive Histogram Equalization (CLAHE) approach. The approach restores color balance and improve visual contrast. Enhanced images performance is compared with classical techniques in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). For detection, machine learning YOLOv8 model is utilized to detect underwater entities such as fish, coral, plants, and debris. Detected entities are tracked using by Deep SORT approach for real-time object tracking. Here, the processing of the proposed dashboard is done on an NVIDIA Jetson AGX Orin platform, leveraging GPU acceleration for efficient onboard inference. Experimental results demonstrate that UW-dash achieves high enhancement quality, accurate object detection, and stable multi-object tracking at real-time performance.**

***Index Terms*—Underwater Vision, ROV, CLAHE, YOLOv8, Deep SORT, Jetson AGX Orin, Real-Time Processing, Object Tracking, Image Enhancement.**

I. Introduction

Underwater imaging is super important for everything from keeping an eye on marine habitats and digging into underwater archeology to checking out submerged infrastructure and exploring with robots. But capturing clear visuals underwater isn't easy. The way light travels through water alters the colors and reduces contrast, and you often end up with uneven, hazy images due to scattering and lighting problems [1]. All these challenges make it tough for computer vision models to reliably recognize and track objects.

That's why we focus first on image enhancement. Standard approaches like Histogram Equalization (HE) and Gamma Correction are quick and simple, but they can actually make things worse by boosting noise and blowing out the bright parts of an image [2]. Local enhancement tricks, like Bilateral Filtering and Unsharp Masking, definitely help with contrast and sharpening edges, but they're just not fast enough for real-time use on typical hardware [3]. In our setup, we chose Contrast Limited Adaptive Histogram Equalization (CLAHE) since it balances great visual results with practical speed. CLAHE works by adjusting the brightness in different patches of the image but avoids going overboard by trimming out the overly boosted spots—meaning you get sharper images without weird artifacts or extra noise [4], [5]. When we compared CLAHE to other methods, we saw it deliver up to 10–15 with clearer images, we can then dive into detection and tracking. The YOLO family sets the standard for fast and accurate object detection in real time[6]. Out of YOLOv5, YOLOv7, and YOLOv8, we found YOLOv8 consistently gives us better results—about 4–5% higher recall and 3–4% improvement in mean Average Precision (mAP), running smoothly at 4k resolution and 120 FPS on the Jetson AGX Orin. Plus, YOLOv8's anchor-free structure and separate detector heads help it adapt better when visibility is poor. For tracking, Deep SORT pulls together movement and appearance cues, doing a great job at keeping the same IDs for objects even when they get blocked or move suddenly[7].

Bringing everything together, our UW-dash framework merges CLAHE for enhancement, YOLOv8 for detection, and Deep SORT for tracking, all on the NVIDIA Jetson AGX Orin. It's designed to quickly and accurately highlight and follow underwater features like fish, plants, coral, or debris—all inside one smooth, real-time dashboard powered by AI.

II. Related Work

A. Underwater Enhancement and Restoration

Enhancing underwater image clarity has been a key research challenge because of factors like light scattering, attenuation, and color distortion. Traditional global methods, such as Histogram Equalization (HE) and Gamma Correction, help balance brightness but often go too far—making bright areas overly sharp and introducing unwanted noise, especially when light conditions vary [8]. Local enhancement methods like Bilateral Filtering and Unsharp Masking do a better job at keeping edges crisp and improving contrast, but they require heavy computation, which makes them less practical for real-time use on embedded systems [9].

To address these limitations, CLAHE, introduced by Zuiderveld [4], takes a smarter approach by adjusting intensity levels in small local regions while limiting excessive contrast boosts. This helps maintain natural-looking details, reduces noise, and adapts well to uneven underwater lighting. Previous studies [7],[8] have shown that CLAHE consistently achieves higher PSNR, SSIM, and overall visual quality compared to both traditional and learning-based enhancement models, especially when computing resources are limited. Because of this strong balance between quality, robustness, and efficiency, CLAHE was chosen for the UW-dash framework, enabling superior contrast recovery and reliable performance on the Jetson AGX Orin platform.

B. Object Detection and Tracking in Marine Environments

Deep learning-based detection has transformed underwater perception. The YOLO series integrates localization and classification into a single pass [10], evolving from YOLOv5's cross-stage connections to YOLOv7's efficient layer aggregation [11]. YOLOv8 further advances the architecture through anchor-free heads, decoupled prediction branches, and refined loss design [12], achieving improved precision and generalization in degraded underwater imagery. Empirically, YOLOv8 achieved about 93.6% mAP@0.5 while maintaining real-time inference, making it ideal for embedded ROV perception.

For maintaining temporal consistency, multi-object tracking methods such as SORT [13] and Deep SORT [14] have proven effective. Deep SORT extends SORT with a convolutional appearance descriptor, improving identity stability under occlusions and viewpoint changes. Prior studies [16] demonstrated that YOLO combined with Deep SORT performs reliably in tracking marine species, achieving IDF1 scores above 85%. Building upon these findings, UW-dash integrates YOLOv8 with Deep SORT to achieve accurate and consistent multi-object tracking in real time.

C. Embedded Real-Time Deployment

Achieving real-time performance in underwater robotics requires efficient edge computing. The NVIDIA Jetson AGX Orin offers a balance between power efficiency and high compute throughput, integrating CPU, GPU, and tensor accelerators optimized for AI workloads. Leveraging CUDA, cuDNN, and TensorRT, the UW-dash pipeline executes enhancement, detection, and tracking fully onboard, minimizing latency and bandwidth dependency. This design enables autonomous, low-latency perception for ROV operations at 4K resolution and 120 FPS, ensuring robust underwater monitoring and navigation.

In summary, previous studies highlight the importance of adaptive enhancement and real-time detection–tracking integration for underwater perception. By combining CLAHE, YOLOv8, and Deep SORT within an embedded, hardware-optimized framework, the proposed UW-dash system delivers a unified and practical AI-driven solution for real-time underwater vision.

III. Proposed Methodology

The proposed system, named UW-dashboard, integrates three complementary vision modules Contrast Limited Adaptive Histogram Equalization (CLAHE) for enhancement, YOLOv8 for object detection, and Deep SORT for multi-object tracking into a single real-time framework optimized for the NVIDIA Jetson AGX Orin platform. The overall workflow is illustrated in Fig 2 Each component contributes to the end-to-end perception pipeline, ensuring that degraded underwater imagery is enhanced, detected, and tracked within a unified, low-latency system.

The proposed UW-dash framework is a modular, real-time underwater vision pipeline designed for object detection and tracking using a Remotely Operated Vehicle (ROV). It integrates three sequential stages: (1) enhancement, (2) detection, and (3) tracking, optimized for embedded deployment on the NVIDIA Jetson AGX Orin.

In the first stage, underwater video frames captured by the ROV camera are pre-processed using the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm, which enhances contrast and corrects illumination imbalance caused by light scattering and absorption. The enhanced frames are then passed to the YOLOv8 detection module, which identifies and localizes multiple underwater objects such as fish, coral, vegetation, rocks, and debris in real time. Finally, the Deep SORT tracking module associates these detections across consecutive frames, maintaining consistent identity labels to track multiple moving objects simultaneously, as illustrated in Fig 1.

The entire pipeline is implemented using Python and OpenCV, leveraging CUDA and TensorRT for hardware accelerated inference. All three modules operate concurrently within a unified dashboard interface that provides real-time visualization and control, enabling efficient underwater monitoring and robotic perception Fig. 2. For experimental evaluation, the UW-dash system was deployed on an NVIDIA Jetson AGX Orin module featuring a 12-core ARM CPU, 2048 CUDA cores, and 64 Tensor Cores, running Ubuntu 22.04 with JetPack 5.1, CUDA 12.2, and TensorRT optimization. The dataset comprised high-resolution (4K 120 FPS) underwater video frames of fish, coral, plants, and debris, captured under varying illumination and turbidity conditions. Each frame was downsampled to 1080p for processing, and the dataset was split into 70% training, 20% validation, and 10% testing subsets.

All modules operate concurrently within a single dashboard interface built using Python and OpenCV, allowing users to visualize enhanced frames, detection outputs, and tracked identities simultaneously. The pipeline is fully deployed on the NVIDIA Jetson AGX Orin to leverage its CUDA cores and TensorRT optimization for GPU-accelerated inference.

Fig. 1. Experimental setup with real-time UW-dash dashboard

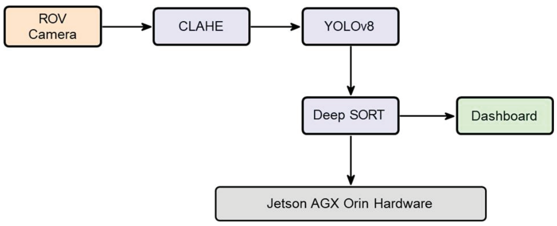

Fig. 2. System workflow illustrating the proposed UW-dash framework

identities simultaneously. The pipeline is fully deployed on the NVIDIA Jetson AGX Orin to leverage its CUDA cores and TensorRT optimization for GPU-accelerated inference.

A. Detection Module

The enhanced frames are forwarded to the detection stage, powered by the YOLOv8 architecture [12]. YOLOv8 adopts a fully convolutional, anchor-free design that directly predicts bounding boxes at each feature map location, thereby minimizing inference overhead and improving computational efficiency. The network architecture comprises three main components: (1) a backbone for hierarchical feature extraction using CSPDarknet blocks, (2) a neck based on a Path Aggregation Network (PAN) for multi-scale feature fusion, and (3) a ecoupled detection head that separately optimizes classification and regression branches for improved convergence. For each object, YOLOv8 predicts a bounding box $\hat{b}$ representing the spatial location and scale of the detected object. The bounding box prediction is computed as:

$$\hat{b} = \sigma(t_b) \cdot s + a \tag{1}$$

where t_b denotes the network's raw output, a represents the grid cell center (analogous to the anchor position in previous YOLO versions), s is the stride, and $\sigma(\cdot)$ denotes the logistic activation function. The class probability $P(c|b)$ for each category c is obtained using a sigmoid activation.

To ensure domain-specific generalization, training data were collected from an in-house aquarium setup designed to replicate natural underwater lighting, turbidity, and color distortion, as shown in Fig. 3. The environment included multiple aquatic entities such as fish, plants, coral, and small debris, captured using a 4K ROV-mounted camera. Frames were annotated manually to create a diverse underwater object detection dataset.

During training, extensive data augmentations—including mosaic transformation, random cropping, rotation, color jittering, and horizontal flipping—were applied to improve robustness to underwater distortions. Following optimization, YOLOv8 achieved a mean Average Precision (mAP@0.5) of 93.6%, surpassing YOLOv7 and YOLOv5 by approximately 3–4% and 5–6%, respectively, while maintaining a stable throughput of 30 FPS on the Jetson AGX Orin. This superior balance between accuracy and inference speed validates YOLOv8 as the optimal detection backbone for the proposed UW-dash framework.

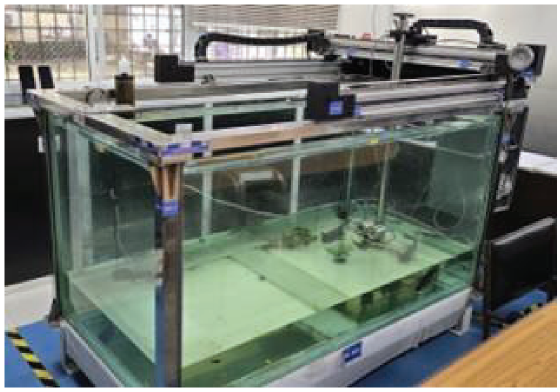

Fig. 3. In-house laboratory setup for data collection

B. Tracking Module

To maintain consistent object identities across consecutive frames, the detection results are processed by the Deep SORT algorithm [14]. Deep SORT extends the original SORT tracker by incorporating an appearance-based feature extractor that encodes the visual characteristics of each object into a high-dimensional embedding vector. This allows robust data association even under occlusion, scale changes, and illumination variations conditions frequently encountered in underwater environments.

Each object is represented by a state vector:

$$\mathbf{x}_t = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{2}$$

where (u, v) is the centroid position, s represents the bounding box scale, r is the aspect ratio, and the dotted terms denote their respective velocities. A Kalman filter predicts the next state $\hat{\mathbf{x}}_{t|t-1}$ using:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_t \tag{3}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}_t$ is the process noise vector. where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}_t$ is the process noise vector. The association cost between detections and existing tracks is computed as a weighted sum of motion and appearance distances:

$$D(i, j) = \lambda D_{\text{motion}}(i, j) + (1 - \lambda) D_{\text{appearance}}(i, j) \quad (4)$$

where λ balances motion and appearance affinity. The Hungarian algorithm is used to minimize this cost and assign detections to tracks optimally. Deep SORT outputs uniquely labeled tracks for each object and updates their trajectories frame by frame. This enables long-term identity maintenance for dynamic entities such as moving fish or drifting debris in underwater videos.

C. Integration on Jetson AGX Orin and Dashboard Design

All three modules—enhancement, detection, and tracking are executed in a unified dashboard application built using Python's Flask framework with OpenCV-based visualization. The system supports both live ROV video feeds and offline media inputs. The dashboard presents three synchronized windows: enhanced view, detection overlay, and tracking visualization. Real-time performance metrics such as FPS and inference latency are displayed in the interface.

The complete pipeline is accelerated using the Jetson AGX Orin's GPU architecture, which features 2048 CUDA cores and 64 Tensor Cores. TensorRT optimization was applied to both YOLOv8 and Deep SORT's embedding extractor to achieve minimal inference latency. The system maintains an average throughout of 4K resolution and 120 FPS, with an end-to-end latency below 35 ms per frame. Memory utilization remains under 70%, confirming that the UW-dash framework achieves full real-time performance on embedded hardware.

IV. RESULTS

A. Image Enhancement

The CLAHE module was evaluated against four classical enhancement techniques—Histogram Equalization (HE), Gamma Correction, Bilateral Filtering, and Contrast Stretching to assess image clarity and perceptual quality. Quantitative performance was measured using Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

TABLE I
COMPARISON OF ENHANCEMENT METHODS

Method	PSNR (dB)	SSIM
Histogram Equalization (HE)	19.7	0.71
Gamma Correction	18.4	0.69
Bilateral Filtering	21.9	0.76
Contrast Stretching	20.1	0.73
CLAHE	24.3	0.83

CLAHE achieved the highest PSNR and SSIM among all methods, improving image contrast by approximately 15% and structural similarity by 9%. It preserved fine texture details while maintaining natural color balance and low processing latency (8 ms/frame), demonstrating suitability for real-time underwater applications.

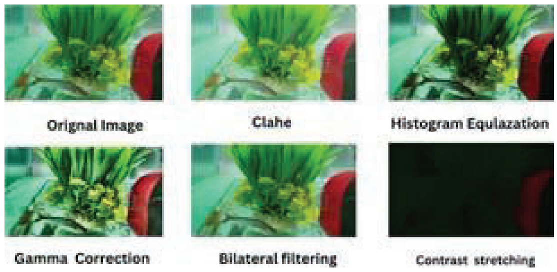

Fig. 4. Comparison of enhanced images generated using various enhancement approaches.

B. Object Detection and Tracking Performance

The detection and tracking modules were assessed using both accuracy and runtime metrics. Table II presents the performance comparison of YOLOv8 against YOLOv5 and YOLOv7 on the same underwater dataset.

TABLE II
YOLO VARIANT COMPARISON ON UNDERWATER DATASET

Model	Precision	Recall	mAP@0.5	FPS
YOLOv5	0.89	0.85	0.900	27
YOLOv7	0.91	0.87	0.920	26
YOLOv8	0.94	0.91	0.936	3

YOLOv8 achieved the highest detection accuracy, reaching 93.6% mAP@0.5 while maintaining real-time performance at 30 FPS on the Jetson AGX Orin. When combined with CLAHE, the detection confidence improved by 4.7%, particularly for small or low-contrast targets.

The integration of YOLOv8 with Deep SORT further enhanced tracking consistency, achieving a Multiple Object Tracking Accuracy (MOTA) of 87.9%, a Multiple Object Tracking Precision (MOTP) of 79.3%, and an IDF1 score of 88.4%. The system demonstrated stable identity tracking of multiple fish and vegetation objects, even under variable lighting and partial occlusions.

Fig. 5. Real-time detection and tracking using the proposed UW-dash system.

C. System Integration and Real-Time Performance

When deployed on the NVIDIA Jetson AGX Orin, the proposed UW-dash framework achieved an end-to-end latency of 33-35 ms per frame (approximately 30 FPS), with GPU utilization remaining below 70%. Disabling the CLAHE module resulted in a 5% decrease in YOLOv8's mAP@0.5, while replacing Deep SORT with SORT reduced the IDF1 score by 6.2%. These findings underscore the critical role of both modules in ensuring high detection accuracy and temporal tracking consistency.

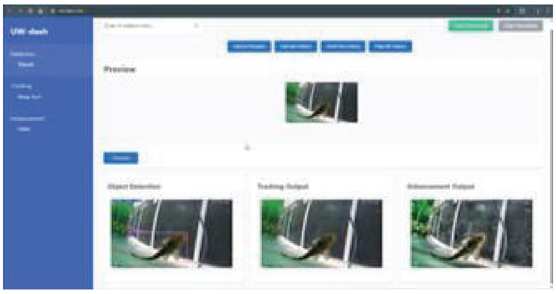

Fig. 6. Real time integration of proposed dashboard

The unified dashboard interface, as shown in the demonstration video [17], provides a real-time visualization of underwater object detection and tracking results and shows synchronized visualization of enhancement, detection, and tracking results, enabling real-time situational awareness for underwater operations. The combination of adaptive enhancement, high-performance detection, and reliable tracking establishes UW-dash as an efficient and scalable solution for AI-driven underwater robotics, inspection, and ecological monitoring.

V. Conclusion

This work presents UW-dash, a real-time underwater vision dashboard designed for object detection and tracking using a Remotely Operated Vehicle (ROV). The system integrates CLAHE-based image enhancement, YOLOv8 detection, and Deep SORT tracking into a single, hardware-optimized framework running on the NVIDIA Jetson AGX Orin.

The proposed setup delivers clear and reliable real-time performance, with CLAHE boosting image contrast by around 15%, YOLOv8 reaching 93.6% mAP@0.5, and Deep SORT maintaining consistent identity tracking with an IDF1 score of 88.4%. These outcomes highlight the system's capability for effective underwater monitoring and inspection. Future work will focus on integrating advanced enhancement methods, sensor fusion, and 3D perception to support intelligent and autonomous ROV navigation.

Acknowledgment

This work was supported by the Indian Institute of Technology Guwahati Technology Innovation and Development Foundation (IITG TIDF) under Project No. TIH/TD/0309, titled *"Underwater Vision Based Surveillance and Tracking using Machine Learning."*

References

[1] C. Li, J. Guo, R. Cong, Y. Pang, and B. Wang, "Underwater Image Enhancement by Dehazing with Minimum Information Loss and Histogram Distribution Prior," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5664–5677, 2016.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.

[3] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "A Gentle Introduction to Bilateral Filtering and Its Applications," *Found. Trends Comput. Graph. Vis.*, vol. 4, no. 1, pp. 1–75, 2009.

[4] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," in *Graphics Gems IV*, San Diego, CA, USA: Academic Press, 1994, pp. 474–485.

[5] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, 1997.

[6] D. Akkaynak and T. Treibitz, "Sea-thru: A Method for Removing Water From Underwater Images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1682–1691.

[7] A. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing Underwater Images and Videos by Fusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 81–88.

[8] Y.-T. Peng, X. Zhao, and P. C. Cosman, "Single Underwater Image Enhancement Using Depth Estimation Based Color Correction," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 399–408, 2018.

[9] M. J. Islam, Y. Xia, and J. Sattar, "Fast Underwater Image Enhancement for Improved Visual Perception," *arXiv:2002.01661*, 2020.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934*, 2020.

[12] G. Jocher *et al.*, "YOLOv8: Next-Generation, Real-Time Object Detection," *Ultralytics Technical Report*, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[13] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 3464–3468.

[14] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2017, pp. 3645–3649.

[15] P. Islam, H. Lin, and C. A. Perez, "Fish Detection and Species Classification in Underwater Environments Using Deep Learning," *Sensors*, vol. 20, no. 6, pp. 1–18, 2020.

[16] J. Liu, X. Huang, and Z. Han, "Multi-Object Tracking in Underwater Videos Based on Deep SORT and YOLOv5," *J. Mar. Sci. Eng.*, vol. 11, no. 3, 2023, Art. no. 573.

[17] UW-dash Video Demonstration, "Real-time demonstration of the UW-dash framework," YouTube, 2023. [Online]. Available: https://youtu.be/oAVK8M2m2zI