iSea: A Domain-Agnostic Dataset for Image Enhancement, Classification, Detection and Segmentation of Subaqueous Specimens

Rushali Sharif Uddin[1], Nusaiba Zaman Manifa[1], Musfirat Hossain[1], Tashfia Zaman[1], Swakkhar Shatabda[1], and Salekul Islam[*]

[1]United International University, Computer Science and Engineering, Dhaka, Bangladesh
[*]corresponding author: Salekul Islam (salekul@cse.uiu.ac.bd)

ABSTRACT

Automated underwater vision has become one of the most significant technologies in accelerating underwater research. The models employed to leverage this task require large datasets with high generalization, limited domain constraints, and proper annotations to tackle the diversified nature of the underwater environment. Existing underwater datasets are domain-constrained and poorly generalized leading to models trained on them failing to detect a myriad of subaqueous specimens. Some datasets are small in size and lack proper annotations for the objects in the image. No single existing dataset caters to all vision-related tasks while considering the aforementioned drawbacks. We propose iSea, a domain-agnostic dataset for image enhancement, classification, detection, and segmentation of underwater images. Our dataset includes 85,093 images from across 4 underwater domains: underwater scene, marine life, wreckage, and garbage, with 35 classes manually annotated after extracting bounding box information using CutLER and validated by our authors using the open-source labeling platform, Roboflow. Masks for multiple instance segmentation and binary segmentation have been generated using CutLER and OpenCV respectively. The images were enhanced using the state-of-the-art enhancement architecture, FUnIE-GAN.

Background & Summary

Underwater vision-focused tasks are of great significance in various domains and has become an important sector of research and development. Automated underwater vision requires little to no human intervention which in turn saves time, and is less expensive and faster. With the recent advancements in this field, it is possible to reach newer depths and gain knowledge about a world that was previously unknown. Leveraging the power of state-of-the-art machine learning architectures and models, underwater exploration has become faster and more accurate. Training these heavy models requires a large-scale dataset composed of diversified images. Underwater vision is characterized by its inherent difficulties. These challenges are poor visibility, varying object appearances, complex backgrounds, environmental variations, the presence of small and camouflaged objects, and inaccurate scale and distance estimation. Besides, factors such as light absorption, scattering, and the presence of suspended particles also affect the performance of the models. Hence the study suffers from a lack of diversified images for analysis. Moreover, the datasets collected and trained for a specific oceanic environment are not suitable for identifying objects from different environments. There also exists a domain gap in the currently available underwater datasets.

Image enhancement, classification, object detection, and segmentation are the most important vision-related tasks in an underwater setting to know the type, location, and behavior of the underwater entities. RUOD[1] is a real-world underwater object detection dataset, which contains 14,000 high-resolution images of 10 object categories that include several challenges like hazy images, color casts, and light interference. However, this dataset cannot be used for training a model for segmentation. The EUVP[2] dataset is a large-scale dataset containing both paired and unpaired instances of poor and good-quality images. It contains 12K paired images and 8K unpaired images to help in perceptual image enhancement and hence there are no object categories included in the dataset. For this reason, the dataset is unsuitable for any tasks other than image enhancement. SUIM[3]is a dataset for the semantic segmentation of underwater images. It is a large-scale dataset comprising 1500 high-resolution images and pixel annotation of 8 object categories but lacks localization information. Hence the dataset is not useful for object detection. On the other hand, the Brackish[4] dataset has 6 object classes, the first dataset comprising images of the temperate brackish waters with varying visibility. It contains 14,518 annotated images captured from videos. While it has a large number of annotated images, the number of classes is very small which does not account for the domain gap that exists in the currently available public underwater datasets. From the existing literature, it is observed that there is a lack of comprehensive representation of the underwater environment in the underwater datasets. Some datasets are catered for image enhancement, some for object detection, and some for segmentation. However, a single dataset cannot be used for all 3 tasks at once. Most datasets address several categories belonging to a specific domain. Some datasets lack high-quality annotation. Manually annotating existing large-scale datasets every time to cater to a specific use is redundant and time-consuming. That is why it is crucial to possess meticulously curated datasets with high-quality annotation for tasks of this nature.

Our aim is to address these limitations in the underwater datasets. That is why we propose a large-scale inclusive dataset without domain constraints containing underwater entities from 4 domains, namely marine life, underwater scene, garbage, and wreckage. The images in our dataset capture several challenges faced in the unpredictable underwater environment. We compiled a total of 85,093 images from 31 domain specific datasets after removing superfluous images. We provide both raw and enhanced images to support the development of image enhancement architectures. For object detection, bounding box information was generated using CutLER[5] for images without annotations. Subsequently, the bounding boxes were fixed and labelled manually by us using Roboflow. We validated each image after annotating them. We generated masks for binary segmentation using OpenCV and CutLER for multi-instance segmentation.

Current research in this field fails to capture the full potential of real-world underwater exploration. Our dataset considers these limitations and can be used as a valuable asset to improve the performance of state-of-the-art models in the underwater context. It can further prove to be beneficial in tasks related to the conservation of marine life and biodiversity.

Methods

To build the iSea dataset we collected around 93 datasets available on the internet. The datasets are individually catered to different computer vision-related tasks like image enhancement, color correction/restoration, single image super-resolution, monocular depth estimation, simultaneous enhancement and super-resolution, image segmentation, salient object detection, and object detection/classification. 62 datasets were removed from consideration due to various factors. We compiled 1,87,053 images across the remaining 31 datasets. Among these images, around 1,01,960 nonessential images were removed. The total number of images in the final compiled dataset summed up to 85,093. The license for each dataset was checked before using it for our work. The entire dataset creation pipeline is primarily divided into three main steps:

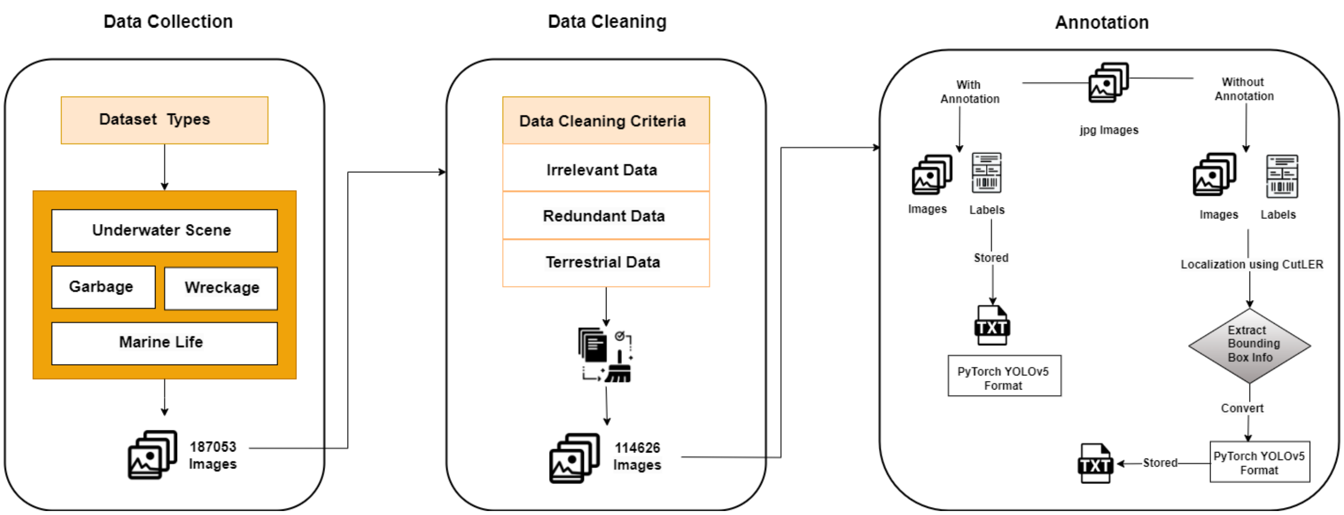


Figure 1. Dataset Creation Pipeline

- **Data Collection**: This step comprises of collecting publicly available datasets from the web.
- **Data Cleaning**: This phase is composed of removal of extraneous images and preparing them for further processing.
- **Annotation**: Object detection coordinates were generated and processed in a suitable format using CutLER in this step.

Data Collection

We attained 93 datasets across the four domains out of which we used 30 datasets. 63 datasets were removed from consideration. The removal is attributed to a range of reasons. The factors leading to the removal of these datasets are presented below:

Reason	Dataset
Site not available	SQUID, Virtual Periscope, Color Restoration, LIACI, MBARI, MOOREA Corals, HABCAM, Oceanwide Dataset
Broken download links	Change Detection, MOUSS, Maris Dataset, Aqualoc, Robot, CADDY
Download forbidden	Fish, Color Correction, SUN Dataset, OUCVision, Fathom Dataset, Coral, Fish4Knowledge, Rockfish, Fish Recognition Data, Fish Trajectory Detection, StellwagenBank, Boulder Imagery, Shipwrecks, NOS-Shipwrecks
Language and image format constraints	Deep Sea Debris Dataset, Aqualife Images, Fish Database, WildFish, Eilat, CoralNet Dataset, Satil
Permission denied from third-party	MUED, NOAA, Diver Detection, Labelled Fishes, USOD
License Issue	Wreck in HETTY TAYLOR, LUCERNE (1873), PRETORIA, WISCONSIN (1881), KATE KELLY (1867), APPOMATTOX (1896), FEDORA (1889), EMPIRE STATE (1862), MOTOR TORPEDO BOAT S 57, Dixie Arrow, KAPITÄN DIEDERICHSEN, USS Monitor, Tamaulipas

Table 1. Rationale for the removal of multiple datasets.

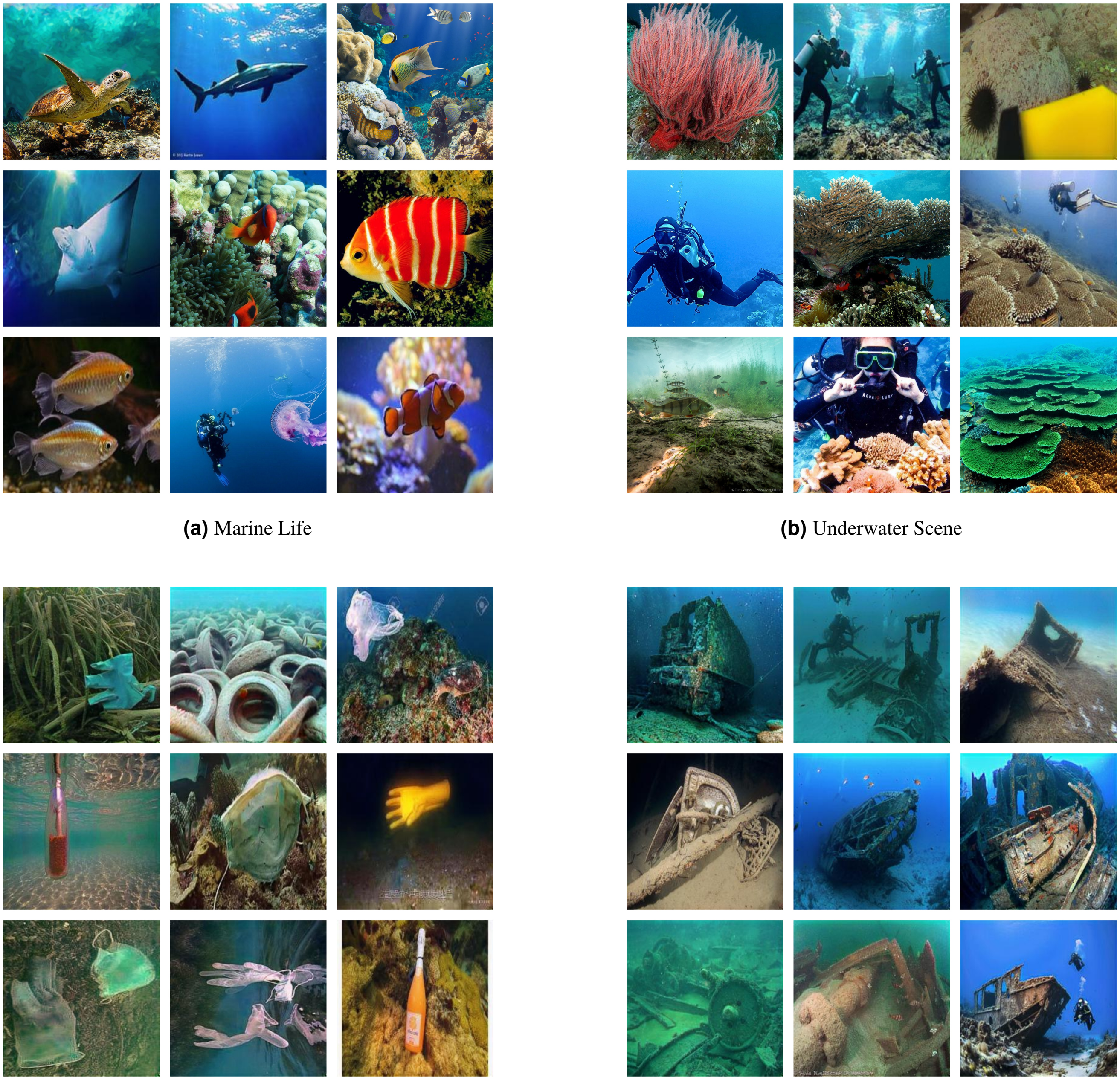
The datasets used in our research come from a variety of sources and are licensed under terms such as the GNU Free Documentation License, Creative Commons, MIT License, Apache License 2.0 and others. These licenses generally allow us to copy, use, and modify the data—as long as we follow the specific conditions set out by each license. In some cases, the use of datasets was explicitly permitted by the original authors, either through statements on the project websites or in the documentation provided within their GitHub repositories. The final set of datasets was restructured to align with three primary tasks in computer vision for the purposes of this research. The main root folder "iSea" contains 3 subfolders named, "objectDetection" , "imageSegmentation" and lastly "imageEnhancement". All the subfolders contain images across the following four domains:

- **Underwater Scene**: This domain is composed of the different entities that make up the underwater environment. The objects in this domain include corals, ROV/AUVs, rocks, divers, aquatic plants, seabed and sea anemones. Unidentified entities are also included in this domain.

- **Marine Life**: This domain refers to the diverse range of living organisms that inhabit beneath the surface of the water. The objects in this domain include molluscan shellfishes, crabs, dolphins, eels, fishes, jellyfishes, lobsters, marine slugs and worms, octopuses, penguins, puffers, seahorses, stingrays, sea urchins, sharks, shrimps, squids, starfishes, turtles, whales, sea cucumbers, echini, sea lions and cuttlefishes.

- **Wreckage**: This domain refers to the remains or debris of ships, boats and aircraft that resulted from an accident or natural disaster. Underwater ruins, statues and artifacts are also included in it.

- **Garbage**: This domain contains images of several types of trash. The objects in these images can be divided into two broad categories: plastic and general trash. Instances with plastic material are labeled as such and all the other types are labeled as trash.



(a) Marine Life

(b) Underwater Scene

(c) Garbage

(d) Wreckage

Figure 2. Different Domains of Underwater Environment

Data Cleaning

The benchmark datasets that we used to create our large-scale image enhancement, object detection, and image segmentation datasets were collected from different sources. The total number of images in the compiled dataset summed up to 1,87,053. However, not all images were usable or relevant to our implementation. We performed data cleaning on the total amount of images. Redundant, irrelevant, and out-of-context images were cleaned. There were several datasets that had images taken from a video. A lot of those images either had no object or the spatial position of the object in consecutive images did not change due to the images being generated every few seconds from a video. Around 1,01,960 images were removed from consideration. Images that allowed us to observe an object from various angles and contexts were retained. Finally, we logged 85,093 images in our dataset.

Annotation

The primary annotation step involves generating localization information and saving it in a text file using the PyTorch YOLOv8 format. Among the images logged in our final dataset, some had annotations while others did not. For annotations we opted for the PyTorch YOLOv5 format due to its popularity and widespread adoption, which ensures compatibility with numerous object detection frameworks. We verified the format for images that already had annotations and converted them into the YOLOv5 format if necessary and saved as '.txt' files. For images without annotations, localization information was generated using CutLER, subsequently converted into the YOLOv5 format, and saved as '.txt' files.

Sl. No.	**Dataset Name**	**Type**	**Raw Images**	**Cleaned Images**	**Bounding Box Annotations**	**Masks/ Mask Annotations**	**Enhanced Images**
1	Underwater Trash Detection	Underwater Scene	7564	218	✓	✗	✗
2	Sea Animals Image Dataset	Marine Life	13711	10916	✗	✗	✗
3	Torpedo Boat Wreck	Wreckage	97	37	✗	✗	✗
4	UW Garbage Debris Dataset	Garbage	4229	1381	✗	✗	✗
5	LIRMM Shipwreck dataset	Wreckage	72	13	✗	✗	✗
6	Shipwreck and Relic Images	Wreckage	41	39	✗	✗	✗
7	Brakish Underwater Dataset[6]	Marine Life	14674	9576	✗	✗	✗
8	Underwater Marine Life Detection Image	Marine Life	4000	4766	✗	✗	✗
9	UTDAC2020[7]	Marine Life	5239	6281	✗	✗	✗
10	DeepFish[8]	Marine Life	4505	4422	✗	✗	✗
11	OzFish[9]	Marine Life	1758	1758	✗	✗	✗
12	DUO	Underwater Scene	7782	6585	✗	✗	✗
13	Trash-Can	Underwater Scene	4762	4780	✗	✗	✗
14	UFO-120	Underwater Scene	3240	1570	✗	✗	✗
15	EUVP	Underwater Scene	7118	15404	✗	✗	✗
16	UGAN	Underwater Scene	5950	1198	✗	✗	✗
17	RUOD[1]	Underwater Scene	13999	2981	✗	✗	✗
18	UOD	Marine Life	105	57	✗	✗	✗
19	URPC-2019	Underwater Scene	4707	4629	✗	✗	✗
20	UIEBD[10]	Underwater Scene	890	846	✗	✗	✗
21	U-45[11]	Underwater Scene	45	45	✗	✗	✗
22	RUIE[12]	Underwater Scene	4227	4137	✗	✗	✗
23	Turbid Dataset	Underwater Scene	15	4	✗	✗	✗
24	OceanDark Dataset	Marine Life	189	156	✗	✗	✗
25	USR248	Marine Life	1302	1294	✗	✗	✗
26	SUIM	Underwater Scene	1625	1622	✗	✗	✗
27	Underwater Plastic Dataset	Garbage	56	13	✗	✗	✗
28	NOAA	Garbage	268	268	✗	✗	✗
29	Pixabay	Garbage	32	32	✗	✗	✗
30	Unsplash	Garbage	73	73	✗	✗	✗
31	**iSea**(our proposed dataset)	**Comprehensive Dataset**	**85,093**	**N/A**	✓	✓	✓

Table 2. Detailed comparison of all source datasets incorporated into the construction of the iSea dataset, including key statistics for iSea

Dataset Preparation

We created three separate datasets. Each dataset is tailored to a specific computer vision task. These datasets combined together, forms the large-scale iSea dataset which serves as a comprehensive resource for various applications in the underwater field. Each dataset was prepared meticulously. Unique requirements of the task the datasets were designed for have been carefully considered. This involved processes like generating enhanced images, annotating images, ensuring proper formatting, generating localization data, validating image quality, and generating masks. We ensured the suitability of the images for training and testing and provided the necessary details to facilitate accurate model development and evaluation. We are offering three separate folders, each designed for a specific computer vision task. They are as follows:

- **Image Enhancement**
- **Object Detection**
- **Image Segmentation**

Image Enhancement

Current image enhancement datasets predominantly consist of images that fail to represent the challenging conditions typically encountered in underwater environments. The datasets often lack the images that account for common underwater challenges such as low-light conditions, blurriness, noise, and color distortion. Consequently, the applicability and robustness of enhancement algorithms trained on such datasets remain limited when confronted with real-world underwater imaging scenarios. Our proposed dataset addresses all these issues. The datasets that we assembled did not have the enhanced version of the images. That is why, we first enhanced the raw images. The enhanced images were generated using a state-of-the-art model named FuNIE-GAN. After generating the images it was divided into train, validation, and test sets. The train set was further divided into two sets: trainA and trainB. The trainA set contains images of the distorted domain and the trainB set contains images of the enhanced domain. Our dataset does not provide one to one mapping of the distorted and enhanced images. It is suitable for unpaired training of image enhancement models. The validation set contains both distorted and enhanced versions of an image.

Object Detection

A significant number of the datasets collected for underwater object detection lack proper annotations. The annotated datasets include only a handful of objects, failing to capture the full range of biodiversity and artifacts present in underwater environments. This domain constraint severely limits the efficacy and generalizability of object detection algorithms, impeding their performance in diverse and complex underwater scenarios. The lack of comprehensive and accurately annotated datasets thus poses a major challenge in the advancement of reliable underwater object detection systems. For object detection, we have images with both annotations and images without annotations. For both types of datasets, we are storing the label with the bounding box information of each entity in a text file for each image. In the case of datasets without annotation, we are first generating bounding boxes using CutLER and then extracting bounding box information from them. However, the bounding box information provided by CutLER is in (x_min, y_min, (x_max, y_max) format. So it was converted in PyTorch YOLOv5 format (x_center, y_center, width, height) and the information was stored in a text file for each image. Lastly, we used Roboflow to annotate the images. Later we validated each image manually. The folder for object detection contains the train, test, and validation folders and each folder has images and labels folder in them.

Image Segmentation

For image segmentation we need mask. We generated the mask using OpenCV. The masks generated by OpenCV are binary masks. The confidence score limit was set to 50% during masks generation. As a result, the images for which confidence score of generated masks was lower than 50% was removed from consideration. So the image segmentation dataset has around **29,786 images**. If an image had several obejcts, a mask image was generated for each object in that image. Those mask images were converted to PNG instance masks with unique pixel values (grayscale) per object. Therefore, each image now had a single mask image mapped to it. Then we converted those PNG instance masks to YOLOv8 polygon format using OpenCV. Masks were also generated by CutLER for multi-class segmentation. The folder for image segmentation contains "multipleInstanceSegmentation" and "binarySegmentation" sub-folders with train, test, and validation; images and masks under each of these folders.

Exploratory Analysis

A comprehensive exploratory analysis is conducted on three datasets provided for image enhancement, image segmentation, and object detection. We have tested our image enhancement architecture on two benchmark metrics, Structural Similarity Index (SSIM) and Peak signal-to-noise Noise Ratio (PSNR), to evaluate our unpaired image enhancement model on a diverse dataset. We have also analyzed Underwater Image Quality Measure (UIQM) which can asses underwater image colorfulness, sharpness, and contrast. We also evaluated the dataset on the underwater image colorfulness measure (UICM), the underwater image sharpness measure (UISM), and the underwater image contrast measure (UIConM).

Analysis of Image Enhancement Dataset

Structural Similarity Index (SSIM)

This metric is based on estimating errors between a distorted image and a reference image honing different powers of the human visual system. It calculates the difference between each corresponding pixel. The structural Similarity Index does that by integrating 3 features:

- Luminance
- Contrast
- Structure

The range of the metric is between -1 and 1 where 1 indicates perfect similarity and -1 represents that the images are very different.

0.0.1 Luminance

Luminance is calculated by averaging all pixel values and is denoted byμ. The formula is given below in Eq.1

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

The luminance comparison function l(x,y) is then a function of μ_x and μ_y

0.0.2 Contrast

It is calculated by taking the standard deviation of all pixels. It is denoted by σ(sigma). The Eq.2

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}} \tag{2}$$

The contrast comparison function c(x,y) is then a function of σ_x and σ_x

0.0.3 Structure

It is computed by dividing the input signal by its respective standard deviation shown in Eq.3

$$(x - \mu_x)/\sigma_x \tag{3}$$

0.0.4 Luminance Comparison Function

It is interpreted using a function l(x,y) where x and y are the two images being compared and μ is the mean of the given image. The comparison function Eq.4

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{4}$$

C1 is a constant to avoid instability when the discriminator becomes 0. It is calculated as shown in Eq.5

$$C_1 = (K_1L)^2 \tag{5}$$

Where L is the dynamic range of pixel values and K1 is normal constants

0.0.5 Contrast Comparison Function

It is denoted by c(x,y) shown in Eq.6 where σ is the standard deviation, and x and y are two images that have to be compared

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{6}$$

C2 is defined in Eq.7,

$$C_2 = (K_2L)^2 \tag{7}$$

Where L is the dynamic range of pixel values and K2 is normal constants.

0.0.6 Structure Comparison Function

It is denoted by s(x,y) defined in Eq.8 where σ is the standard deviation, and x and y are two images that have to be compared

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{8}$$

and σ_{xy} in Eq.9 is defined by,

$$\sigma_{xy} = \left(\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y) \right) \tag{9}$$

0.0.7 The SSIM Scoring Criteria

The SSIM Scoring Scheme is shown in Eq.10

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{10}$$

where $\alpha > 0$, $\beta > 0$, $\gamma > 0$ denote the relative importance of each of the metrics. To simplify the expression depicted in Eq.11, if we assume, $\alpha = \beta = \gamma = 1$ and C3 = C2/2, we can get,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{11}$$

0.0.8 Peak signal-to-noise Noise Ratio (PSNR)

The PSNR evaluates the reconstruction quality of a generated image x compared to its ground truth y based on their Mean Squared Error (MSE) shown in Eq.12. The typical value of PSNR ranges from 30 to 50 dB. Values exceeding 40 are considered very good whereas below 20 dB is unacceptable.

$$PSNR(x,y) = 10 log_{10} \left[255^2 / MSE(x,y) \right] \tag{12}$$

0.0.9 Underwater Image Quality Measure (UIQM)

The UIQM comprises three underwater image attribute measures: the underwater image colorfulness measure (UICM), the underwater image sharpness measure (UISM), and the underwater image contrast measure (UIConM)

Insights: The iSea dataset is notable as a domain-agnostic benchmark because its image quality and enhancement scores are considerably lower than those of other underwater datasets. Although its SSIM (0.87) and PSNR (31.55) are at moderate levels, its UICM (3.11), UISM (5.55), UIConM (0.26), and UIQM (2.67) rank among the lowest overall. These metrics indicate that iSea comprises highly diverse, unfiltered, and challenging visual material, capturing a variety of underwater settings, including murky waters, fluctuating lighting conditions, and various themes such as marine life, shipwrecks, and debris. In contrast to domain-specific datasets like EUVP, DeepFish, or RUIE, which typically showcase cleaner or enhanced visuals, iSea provides a more authentic and varied dataset that is more appropriate for assessing the robustness and generalization capabilities of image enhancement and object detection models. The low scores reflect real-world conditions rather than indicating poor quality, which makes iSea a valuable asset for training and benchmarking models designed for diverse and unconstrained underwater scenarios.

Analysis of Object Detection Dataset

0.1 Entropy-Based Diversity

Entropy measures how uniformly the data are distributed between classes and helps quantify the diversity of images between classes. High entropy indicates a balanced class distribution, which is critical for domain-agnostic learning. A model trained on a balanced dataset is less likely to overfit to dominant classes typical of one domain and is more likely to generalize across domains. High entropy means balanced class representation. In contrast, data sets with low entropy can lead to biased models, as they disproportionately learn patterns from overrepresented classes that potentially underperform in underrepresented ones. Measures how uniformly data is distributed across classes and helps detect class imbalance.
Insights: iSea has a high entropy that indicates balanced class distribution, which is good for generalization.

0.2 Unique Label Ratio

The unique label ratio is calculated as the number of classes in a dataset divided by the total expected number of classes (in this case, 35). This metric provides insight into how comprehensively the dataset covers the expected label space. A high unique label ratio indicates broad coverage of the class taxonomy, making the dataset more versatile for tasks requiring diverse class representation. On the other hand, a low unique label ratio might suggest missing or underrepresented labels, limiting the model's ability to learn about the entire domain and reducing its applicability in real-world scenarios. A dataset that includes all or most of the expected classes is more domain-agnostic because it's not tailored to a subset of real-world categories. Missing labels suggest blind spots in certain domains.
Insights: iSea has a perfect score which means all 35 classes are present.

0.3 Total Images

This metric represents the total number of images in the dataset, giving a measure of its size. Larger datasets typically contain more variability, which is essential for training robust models, particularly deep learning models that require substantial data to avoid overfitting. It shows the volume of the dataset and helps to gauge whether there's enough data to train a deep learning model. Large datasets generally offer greater intra-class diversity (e.g., different environments, angles, lighting). More images increase the chances of including examples from multiple domains, which eventually supports models to generalize better.
Insights: iSea is excellent for training deep learning models.

0.4 Mean Images per Class

The mean images per class measures the average number of images available for each class in the dataset. This metric provides a sense of how well-represented each class is, on average. A high mean suggests that, on average, classes have sufficient examples for the model to learn from. However, if the mean is low, it may indicate that some classes are underrepresented, which could lead to poor generalization for those categories. A high mean alone does not guarantee balance; it must be interpreted with other metrics like standard deviation.
Insights: iSea has a good representation on average.

0.5 Standard Deviation of Images per Class

The standard deviation of images per class measures the variability in the number of images across different classes. A low standard deviation indicates that the classes have roughly the same number of images, suggesting a balanced dataset. Conversely, a high standard deviation points to significant differences in class representation, highlighting potential class imbalance. Class imbalance can lead to biased models that overfit to dominant classes while underperforming on minority classes. Thus, managing standard deviation is critical for building fair and unbiased models.
Insights: iSea has a high standard deviation that indicates large variance across classes. So before using this dataset, it may require class balancing strategies or subsampling from dominant classes.

0.6 Median Images per Class

The median is the middle value in the distribution of images per class. Unlike the mean, it is less affected by outliers or extreme values. A high median suggests that most classes have a decent number of images, while a low median might indicate that many classes are underrepresented. If the median is significantly lower than the mean, it can highlight that a few classes dominate the dataset, skewing its balance. The median provides a robust measure to understand the central tendency of class representation. A low median relative to the mean could mean many underrepresented classes, which reduces cross-domain performance, especially on long-tail or rare classes.
Insights: The median number of images per class is 1,933 — indicating that while the distribution is skewed, a majority of classes are still well-represented. The gap between the mean and median suggests the presence of dominant classes, which we address by recommending loss-balancing strategies during training.

0.7 Mode Images per Class

The mode represents the most frequently occurring number of images per class. It shows the "typical" class size in the dataset. If the mode is high and close to the mean and median, it suggests that the dataset is balanced and that many classes have a similar number of images. A low mode, especially if it is far from the mean and median, might indicate that several classes have very few examples, highlighting imbalance. Understanding the mode helps identify whether the dataset has a consistent pattern of class representation or whether certain classes deviate significantly from the norm.
Insights: Many classes might have few examples; underrepresentation possible.

Sl. No.	Dataset Name	Image Enhancement Quality Metrics						Object Detection Quality Metrics						
		SSIM	PSNR	UICM	UISM	UIConM	UIQM	Entropy Diversity	Unique Label Ratio	Total Images	Mean Img/Cls	STD Img/Cls	Median Img/Cls	Mode Img/Cls
1	Underwater Trash Detection	0.92	33.16	14.29	116.96	-3.81	62.00	2.49	0.46	218	3.56	5.30	0.00	0.00
2	Sea Animals Image Dataset	0.96	34.99	11.48	121.43	-11.71	61.81	4.33	0.60	10916	983.19	332.15	917.00	917.00
3	Torpedo Boat Wreck	0.83	29.86	14.72	124.45	-2.91	66.06	NaN	0.03	37	0.00	0.00	0.00	0.00
4	UW Garbage Debris Dataset	0.82	28.95	9.83	121.76	-8.08	62.21	1.22	0.43	1381	0.60	1.54	0.00	0.00
5	LIRMM Shipwreck dataset	0.70	21.17	-3.30	123.06	0.28	60.59	NaN	0.09	13	0.00	0.00	0.00	0.00
6	Shipwreck and Relic Images	0.77	27.28	7.53	126.03	-2.15	64.84	NaN	0.17	39	0.00	0.00	0.00	0.00
7	Brackish Underwater Dataset[6]	0.94	30.39	11.70	119.64	-2.40	62.85	NaN	0.03	9576	0.00	0.00	0.00	0.00
8	Underwater Marine Life Detection Image	0.92	31.31	12.56	126.61	-14.04	64.26	2.20	0.43	4766	84.53	155.14	6.00	0.00
9	UTDAC2020[7]	0.84	32.74	11.45	123.03	-4.92	63.96	1.29	0.37	6281	341.62	771.56	0.00	0.00
10	DeepFish[8]	0.87	31.95	14.10	120.17	-5.41	63.23	NaN	0.03	4422	0.00	0.00	0.00	0.00
11	OzFish[9]	0.70	28.00	5.18	117.88	-4.77	59.54	NaN	0.03	1758	0.00	0.00	0.00	0.00
12	DUO	0.83	32.62	9.99	122.98	-4.20	63.65	0.84	0.46	6585	1566.25	4846.94	0.00	0.00
13	Trash-Can	0.90	33.32	12.83	110.84	-3.61	58.54	2.16	0.46	4780	68.56	119.90	1.50	0.00
14	UFO-120	0.87	30.88	13.08	124.69	-12.78	63.71	0.97	0.66	1570	151.48	605.97	6.00	0.00
15	EUVP	0.89	31.07	12.66	124.87	-11.54	63.92	2.81	0.94	15404	877.52	2646.24	221.00	0.00
16	UGAN	0.85	31.29	10.80	125.63	-10.93	63.87	0.90	0.51	1198	38.33	124.84	0.00	0.00
17	RUOD[1]	0.84	30.19	11.80	122.06	-11.15	62.33	2.57	0.80	2981	479.93	1266.98	33.50	0.00
18	UOD	0.77	29.49	11.05	124.83	-10.40	63.65	NaN	0.23	57	0.00	0.00	0.00	0.00
19	URPC-2019	0.83	33.24	10.86	121.87	-3.18	63.56	NaN	0.03	4629	0.00	0.00	0.00	0.00
20	UIEBD[10]	0.82	29.49	10.87	124.64	-7.93	63.99	0.72	0.34	846	0.42	1.11	0.00	0.00
21	U-45[11]	0.73	27.69	7.78	125.99	-2.89	64.75	1.71	0.34	45	1.58	2.78	0.00	0.00
22	RUIE[12]	0.83	33.72	13.73	123.83	-3.37	65.36	NaN	0.03	4137	0.00	0.00	0.00	0.00
23	Turbid Dataset	N/A	N/A	N/A	N/A	N/A	N/A	NaN	0.06	4	0.00	0.00	0.00	0.00
24	OceanDark Dataset	0.90	32.67	15.85	119.16	-5.43	63.25	0.00	0.23	156	24.25	64.16	0.00	0.00
25	USR248	0.86	30.93	13.19	125.62	-11.97	64.37	1.82	0.74	1294	66.42	233.61	9.50	0.00
26	SUIM	0.84	30.36	11.38	121.65	-10.53	62.13	1.81	0.71	1622	103.56	347.70	9.00	0.00
27	Underwater Plastic Dataset	N/A	N/A	N/A	N/A	N/A	N/A	NaN	0.14	5	0.00	0.00	0.00	0.00
28	NOAA	0.77	27.43	7.92	115.93	-5.85	59.17	0.00	0.23	268	0.25	0.66	0.00	0.00
29	Pixabay	0.91	33.10	9.53	102.09	-2.01	53.50	NaN	0.20	32	0.00	0.00	0.00	0.00
30	Unsplash	0.81	30.62	8.17	119.49	-10.19	60.16	NaN	0.17	73	0.00	0.00	0.00	0.00
31	**iSea** (our proposed dataset)	**0.87**	**31.55**	**3.11**	**5.55**	**0.26**	**2.67**	**3.70**	**1.00**	**85093**	**6335.51**	**12937.12**	**1933.00**	**26.00**

Table 3. Comparison of image enhancement and object detection metrics for iSea and its benchmark datasets.

Class Distribution Statistics — iSea Dataset

Insights: Class distribution in iSea exhibits a long-tailed pattern — a common trait in real-world datasets. While the average class has over 6,000 images, the mode is just 26, and the median sits at 1,933, indicating that many classes are underrepresented relative to a few dominant ones. This is further reflected in the high standard deviation (12,937). Such a distribution mirrors natural environments, where some object types (e.g., fish, coral) are abundant and others (e.g., torpedoes, submarines, microplastics) are inherently rare. This makes iSea an ideal benchmark for long-tail learning, zero-shot generalization, and real-world robustness in object detection models

Statistic	Value	Interpretation
Number of Classes	35	Full label coverage — complete class taxonomy
Total Images	85,093	Large-scale dataset
Mean Images per Class	6,335.51	On average, each class has a strong representation
Median Images per Class	1,933	Half the classes have fewer than 1,933 images — suggests skewed distribution
Mode Images per Class	26	The most frequent class size is very small, indicating a long tail of underrepresented classes
Standard Deviation	12,937.12	Very high spread — large disparity between most- and least-represented classes
Min Class Size (estimated)	26 (from mode)	Some classes are sparsely populated
Max Class Size (estimated)	Possibly >20,000 (due to std dev)	Suggests a few classes are dominant in size
Skewness (implied)	Right-skewed / Long-tail	A few large classes, many small ones

Table 4. Image enhancement metric comparison of iSea with some other underwater datasets.

Data Records

A brief overview of the datasets that were used to create our large-scale dataset are detailed in **??**. **The iSea dataset has been made available for public download through Provide the link. The data can be downloaded without any need for registration. The total size of our dataset is 323 MB. The folder structure containing the dataset with all relevant files is described below.**

Folder Structure:

The iSea folder contains 3 folders. The root folder for the image segmentation dataset is named imageSegmentation. The image segmentation root folder has two sub-folders "images" and "labels" and a YAML file. These two folders further contain three sub-folders "train", "valid" and "test" containing the train, validation and test set images and their annotations respectively. The root folder for the object detection dataset is named objectDetection and has three sub-folders named "train", "valid" and "test" containing the train, validation and test sets respectively. Similar to image segmentation dataset, each sub-folders in the object detection dataset has two more sub-folders named "images" and "labels". The images for object detection are in the "images" folder. The corresponding annotations of those images are in the "labels" folder. All annotations are in the PyTorch YOLOv5 format. The description of the classes is given in the "classes.txt" file. Lastly, the dataset for image enhancement has a root folder named imageEnhancement with folders "trainA", "trainB", "validation", and "test" in it. The folder "trainA" contains distorted images, and "trainB" has the enhanced domain without any one-to-one mapping. The "validation" folder has a mix of distorted and enhanced images and the "test" folder contains distorted images that need to be enhanced.

Technical Validation

Our authors reviewed the annotations and masks that were generated for each image. To test our dataset's validity, we trained localization and segmentation using YOLOv8n and YOLOv8n-seg respectively, and FUnIE-GAN was used for image enhancement. The object detection dataset consisted of approximately 87,000 images and the segmentation dataset consists of around 30,000 images. The training was done in Kaggle. Both the models were trained for 10 epochs.

Task	Model	Type	Precision	Recall	mAP@0.5
Localization	YOLO8n	Box	0.531	0.476	0.467
Segmentation	YOLO8n-seg	Polygon	0.597	0.383	0.399

Table 5. Performance comparison of YOLO models

(a) Detection of a Coral with a confidence score of 0.51 **(b)** Detection of a Stingray with a confidence score of 0.46 **(c)** Detection of a Seahorse with a confidence score of 0.41 **(d)** Detection of an Equipment with a confidence score of 0.98

Figure 3. Example images illustrating the model's detection performance on test data.

(a) Segmentation of a crab **(b)** Segmentation of an Eel **(c)** Segmentation of a Snake **(d)** Segmentation of Dolphins

Figure 4. Example images illustrating the model's segmentation performance on test data.

Usage Notes

The dataset is made commercially available for any purpose. The images are free to copy, share, or redistribute in any format or medium. It can also be adapted, transformed, remixed, and built upon. The dataset is licensed under a Creative Commons "Attribution 4.0 International" license (https:// creativecommons.org/licenses/by/4.0/). Any sort of work that uses this dataset is encouraged to cite the paper. The iSea folder contains separate subfolders for each computer vision-related task for easy access. These subfolders can be easily used upon downloading the iSea folder. **(Provide ReadMe)**

Code availability

The steps that we followed to instantiate CutLER and FUnIE-GAN are briefly described here **(link to the README file with instructions)**. All the annotations in our dataset are in the PyTorch YoloV5 Format. Datasets with already existing annotations were converted to the PyTorch Yolov5 format using the code provided at **(Give Github link)**. Bounding boxes and multiple instance masks were generated using CutLER for datasets without annotations. The bounding box information generated by CutLER was converted into PyTorch yolov5 format **(the code)** and was stored in a text file for each image. The labels for each bounding box were generated using a Python script **Code here** and then manually validated **Code here**. The whole process is described broadly in the Methods section. The masks **code here**. The masks are generated using CutLER. The enhanced version of the images were generated using the state-of-the-art enhancement model FUnIE-GAN.**(Provide the evaluation codes after it is fixed. Include Technical Validation codes.)**

References

1. Chenping Fu, Risheng Liu, Xin Fan, Puyang Chen, Hao Fu, Wanqi Yuan, Ming Zhu, Zhongxuan Luo. Rethinking general underwater object detection: Datasets, challenges, and solutions. https://github.com/dlut-dimt/RUOD.

2. of Minnesota, U. Euvp. https://drive.google.com/drive/folders/1ZEql33CajGfHHzPe1vFxUFCMcP0YbZb3 (2024). Visited on 2024-1-25.

3. of Minnesota, U. Suim. https://irvlab.cs.umn.edu/resources/suim-dataset (2024). Visited on 2024-1-25.

4. Pedersen, M., Haurum, J. B., Gade, R., Moeslund, T. B. & Madsen, N. Detection of marine animals in a new underwater dataset with varying visibility. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2019).

5. Wang, X., Girdhar, R., Yu, S. X. & Misra, I. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3124–3134 (2023).

6. Pedersen, M., Lehotský, D., Nikolov, I. & Moeslund, T. B. Brackishmot: The brackish multi-object tracking dataset. In *Computer Analysis of Images and Patterns (CAIP 2022)*, vol. 13509 of *Lecture Notes in Computer Science*, 17–28, 10.1007/978-3-031-31435-3_2 (Springer, 2022).

7. SRMIST. Utdac2020 dataset. https://universe.roboflow.com/srmist-vx65l/utdac2020-kkoqh (2023). Visited on 2024-01-25.

8. Al Muksit, A. *et al.* Yolo-fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecol. Informatics* **72**, 101847 (2022).

9. Al Muksit, A. *et al.* Yolo-fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecol. Informatics* **72**, 101847 (2022).

10. Li, C. *et al.* An underwater image enhancement benchmark dataset and beyond. https://drive.google.com/file/d/12W_kkblc2Vryb9zHQ6BfGQ_NKUfXYk13/view (2019).

11. Li, H., Li, J. & Wang, W. A fusion adversarial underwater image enhancement network with a public test dataset. arxiv 2019. https://github.com/IPNUISTlegal/underwater-test-dataset-U45-.

12. Liu, R., Fan, X., Zhu, M., Hou, M. & Luo, Z. Real-world underwater enhancement: challenges, benchmarks, and solutions.

13. Kaufman, D. *et al.* A global database of holocene paleotemperature records. *Sci. Data* **7**, 115, https://doi.org/10.1038/s41597-020-0445-3 (2020).

14. Figueredo, A. J. & Wolf, P. S. A. Assortative pairing and life history strategy – a cross-cultural study. *Hum. Nat.* **20**, 317–330, https://doi.org/10.1007/s12110-009-9068-2 (2009).

15. Babichev, S. A., Ries, J. & Lvovsky, A. I. Quantum scissors: teleportation of single-mode optical states by means of a nonlocal single photon. Preprint at https://arxiv.org/abs/quant-ph/0208066 (2002).

16. Behringer, R. *Manipulating the mouse embryo: a laboratory manual* (Cold Spring Harbor Laboratory Press, New York, 2014).

LaTeX formats citations and references automatically using the bibliography records in your .bib file, which you can edit via the project menu. Use the cite command for an inline citation, e.g.[13–16]. For data citations of datasets uploaded to e.g. *figshare*, please use the `howpublished` option in the bib entry to specify the platform and the link, as in the `Hao:gidmaps:2014` example in the sample bibliography file. For journal articles, DOIs should be included for works in press that do not yet have volume or page numbers. For other journal articles, DOIs should be included uniformly for all articles or not at all. We recommend that you encode all DOIs in your bibtex database as full URLs, e.g. https://doi.org/10.1007/s12110-009-9068-2.

Acknowledgements

We would like to express our gratitude to Dr. Salekul Islam for the creative liberty and continuous help along the course of this project. The research is supported by the Institute for Advanced Research (IAR) Publication Grant of United International University (UIU), Dhaka, Bangladesh.

Author contributions statement

R.S.U. and N.Z.M. generated the database and retrieved the majority of images. All authors cleaned the data. R.S.U. helped in the exploratory analysis of the database. All authors annotated and validated the annotations. R.S.U. did the technical analysis for the dataset. R.S.U. generated the bounding box information using CutLER and masks for segmentation. R.S.U. and N.Z.M. drafted the paper. S.S. and S.I. supervised the project and helped actively in manuscript preparation.

Competing interests

The authors declare no competing interests.

Figures & Tables

Figures, tables, and their legends, should be included at the end of the document. Figures and tables can be referenced in LaTeX using the ref command, e.g. Figure **??** and Table 1.

Authors are encouraged to provide one or more tables that provide basic information on the main 'inputs' to the study (e.g. samples, participants, or information sources) and the main data outputs of the study. Tables in the manuscript should generally not be used to present primary data (i.e. measurements). Tables containing primary data should be submitted to an appropriate data repository.

Tables may be provided within the LaTeX document or as separate files (tab-delimited text or Excel files). Legends, where needed, should be included here. Generally, a Data Descriptor should have fewer than ten Tables, but more may be allowed when needed. Tables may be of any size, but only Tables which fit onto a single printed page will be included in the PDF version of the article (up to a maximum of three).

Due to typesetting constraints, tables that do not fit onto a single A4 page cannot be included in the PDF version of the article and will be made available in the online version only. Any such tables must be labelled in the text as 'Online-only' tables and numbered separately from the main table list e.g. 'Table 1, Table 2, Online-only Table 1' etc.