ISSN: 2584-0495 Vol. 3, Issue 3, pp. 1605-1613

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

A Comprehensive Review of Deep Learning Algorithms for Underwater Trash Detection: Advancements, Challenges, and Future Directions

Jyoti Sandur, Alka Barhatte

Cite as: Sandur, J., & Barhatte, A. (2025). A Comprehensive Review of Deep Learning Algorithms for Underwater Trash Detection: Advancements, Challenges, and Future Directions. International Journal of Microsystems and IoT, 3(3), 1605–1613. https://doi.org/10.5281/zenodo.18146321

Published online: 10 March 2025

https://doi.org/10.5281/zenodo.18146321

International Journal of Microsystems and IoT
Vol.3, Issue 3, pp.1605-1613; DOI: https://doi.org/10.5281/zenodo.18146321

A Comprehensive Review of Deep Learning Algorithms for Underwater Trash Detection: Advancements, Challenges, and Future Directions

Jyoti Sandur, Alka Barhatte

MIT WPU Electrical and Electronics Engineering, Bharati Vidyapeeth (Deemed to be University) Pune, India

ABSTRACT

Underwater pollution, particularly from plastic and other debris, poses a serious environmental threat to marine ecosystems and biodiversity. Traditional methods for detecting underwater trash, such as sonar-based systems and manual inspections, face significant limitations, especially in deep, turbid waters with low visibility. Recently, deep learning algorithms, including Convolutional Neural Networks (CNNs) and frameworks like YOLO (You Only Look Once) and Faster R-CNN, have shown promising results in automating underwater trash detection. These models, trained on large datasets like Trash Can, offer high accuracy and real-time detection capabilities. However, challenges persist, such as environmental variability, including changes in water clarity, light conditions, and surface disturbances, which can distort images and reduce detection accuracy. Additionally, the lack of comprehensive, annotated datasets, particularly for small debris like microplastics, and issues related to data imbalance complicate the development of robust detection systems. Despite these obstacles, deep learning models continue to improve with advancements in model architectures, data augmentation techniques, and integration of multimodal sensor data, such as sonar, to enhance detection in varied underwater conditions. The future of underwater trash detection lies in overcoming these challenges by optimizing lightweight, real-time models for resource-constrained platforms and enhancing detection of small and overlapping debris. This paper provides a comprehensive review of current deep learning techniques for underwater trash detection, highlighting advancements, challenges, and future research directions for improving model performance and scalability.

KEYWORDS

underwater trash detection, deep learning, YOLO, Faster R-CNN, CNN, microplastics, environmental variability, data augmentation, real-time detection, marine pollution.

1. INTRODUCTION

Marine pollution, and the increasing prevalence of trash underwater, is one of the most serious environmental challenges today. The rapid accumulation of plastics and debris in aquatic ecosystems represents a clear threat to marine biodiversity, disrupts ecological systems, and threatens the health of aquatic organisms. Plastics will remain in marine environments for a long time due to their durability, thereby inducing long-term ecological effects. The annual estimate of 8 million metric tonnes of plastic trash entering the oceans continues to add to the concerns of marine pollution [1]. Marine debris—in particular plastics—seriously impacts marine species through ingestion, entanglement, and disruption of habitat. Plastic waste is claimed to kill millions of marine animals every year. And, when it enters food chains, it can negatively affect human health [2]. Solving this challenge requires efficient and real-time identification and classification systems that detect and classify marine debris. Efficient and accurate identification of marine debris is crucial for conservation efforts that promote and protect marine habitats, as well as minimizing impact and harm to the environment.

Fig. 1 Underwater Trash Detection in Marine Environments [3]

Conventional methods for monitoring marine debris, including sonar-based systems and physical inspections of sites, come with considerable limitations given lower visibility in deep or turbid waters. These methods usually do not have the scalability or accuracy available for operations in real-time or on a larger computational scale. To this end, deep learning techniques have been an intriguing method to automatically search for debris. Convolutional neural networks (CNNs) and advanced systems such as YOLO (You Only Look Once) or Faster R-CNN have made a lot of progress in works involving locating objects and classifying [4].

There are some problems with using deep learning to find debris underwater, even though these models had high mean average precision (mAP) scores and real-time detection speeds when tested on larger datasets like the Trash Can dataset. They will probably work well for keeping an eye on trash in underwater environments. The underwater environment is variable by nature; environmental features may alter water clarity, change lighting, and disturb the surface (e.g proximity to boat traffic or weather), leading to distortion in the images and reduced accuracy in detection models. In addition to limitations of the underwater system, the availability of data, specifically for micro plastics, and data imbalance, where categories of debris can be frequently larger than other types, will complicate the process of building strong and accurate models [5]. These limitations hinder the generalization of deep learning models across diverse underwater environments and further complicate the detection of rare debris types.

Even with these barriers, deep learning models continue to show great potential for underwater trash detection. The future will involve working to overcome these shortcomings using advanced data augmentation, improved model architectures, and multimodal sensor data, like sonar, to assure detection accuracy in all types of environments [6].

Furthermore, improvements in real-time processing systems and computational efficiencies will be pivotal to ensure these models operate successfully in the marine environments in the real world. This paper provided a complete summary of the most recent deep learning models in detecting underwater debris and shows advancements in the field with the challenges that remain to make detection more accurate and scalable in the future.

2. Deep Learning Algorithms for Underwater Trash Detection

One area of neural networks that gets a lot of attention is deep learning. You can think of neural networks as the building blocks of deep learning systems. A "deep" neural network is one that has more than three layers of nodes [7].

A deep neural network is put together in Figure 2. Some of the levels are hidden, and there is only one exit layer. Most of the time, deep neural networks work in a way called "feed-forward." This means that the data only moves from the entry layer to the exit layer. It is also possible for data to go from the output layer to the input layer and back again. This process is known as back spread. When we use back propagation to train the deep learning model, we can find the mistake in every cell. We can now change the way we do things to get better results.

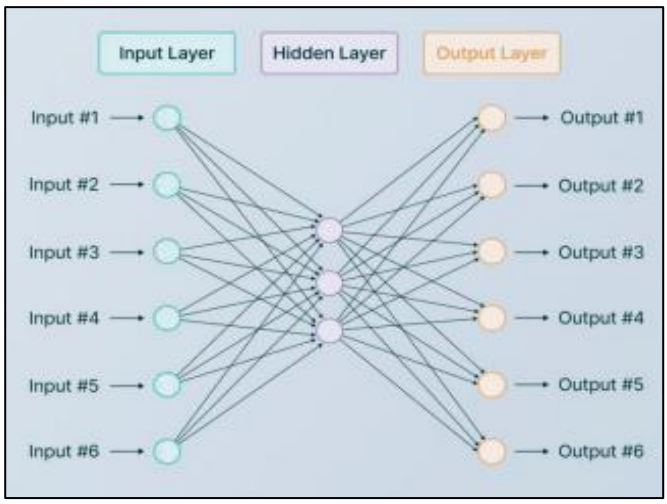

Fig 2. Neural Network Architecture [8]

In Figure 2, you can see how the Neural Network works. This network is made up of three main parts. There is an output layer, a secret area, and an entry area.

- **Input layer:** The first neural layer's input layer is in charge of bringing in the first data so that it can be processed by the layers that follow.
- **Hidden layer:** This is the second kind of layer, and there may be one or more of them in neural networks for high efficiency and complexity. They do many things at once, like changing data and making features automatically. After that, the information is sent to the next layer to be handled further.
- **Output layer**: In the last step, projections are found that meet the needs. Many layers of nodes are connected to each other to make up an artificial neural network. These layers are called "node layers," and they work a lot like the brain's neural network.

As you can see in Figure 2.1, a neural network usually has one input layer, one or more hidden layers, and one output layer. No matter how the program is built, the number of secret layers may change. The number of input and output levels stays the same.

Each of the linked nodes, which are also called neurones, has a weight and a cutoff that go with it. When the output value of a neurone is higher than a certain cutoff number, that neurone is turned on. When a neurone is triggered, it sends data to the next layer of the network [18]. If the chosen neural network has more than one working layer, the steps above are done more than once to get a single result. The results from the processing stages are used as sources to figure out the neural network's end output. They change the weights of the neurons in this step to make the neural network as accurate as possible. The buried layers are also known as processing layers. The secret layers then handle the collected data in this way to get correct features or classifications.

1. YOLO

YOLO, which stands for "You Only Look Once," is one of the best known deep learning models for finding things in real

time." Because of how it's built, YOLO can identify different kinds of objects in a single pass over a picture. This means that it can be used for real-time analysis and is useful for any quick task, like finding trash underwater. From YOLOv3 to YOLOv4 and now to YOLOv8, YOLO has gotten faster, more accurate, and better at finding items of different sizes. This makes it the best choice for real-time recognition in tough settings like seas and marine ecosystems.

YOLOv3 (2018)

YOLOv3 was a major improvement over previous versions, contributing to improved speed and accuracy in the detection. An important element of YOLOv3 was adopting a deeper network architecture with residual connections that improved accuracy at feature extraction. Multi-scale forecasts were also used by YOLOv3 to find items of different sizes and forms that were moving through a single picture.These capabilities are especially pertinent in underwater environments, where different kinds of debris, such as plastic waste can be identified despite challenges like changing levels of water clarity and light refractions [9].

YOLOv4 (2020)

YOLOv4 was launched in 2020 and was designed with several optimizations to better use performance on both GPU and CPU systems to fit a wider array of hardware. To make it easier to find, YOLOv4 used methods like the Mish activation function, weighted leftover links, and the CSPDarknet53 backbone.The model was particularly optimized for use on large datasets, allowing it to process high-resolution images in real-time. YOLOv4's ability to perform accurate object detection while maintaining computational efficiency made it ideal for underwater trash detection, where large volumes of environmental data need to be processed rapidly. YOLOv4's versatility in both hardware configurations and detection performance marked a significant step forward [10] .

Overview of YOLOv5

The new way we find targets underwater is based on what we talked about when we talked about the general structure or model structure of YOLOv5. In 2020, Glenn Jocher put out this record. YOLOv5 adds to the model framework of the YOLO series programs that came before it.

1. Proposed Model

This section talks about the better YOLOv5 underwater target recognition method. Figure 3 shows that we started by processing the data, which meant that we cleaned it up and gave it names. After that, the better YOLOv5 network was used to increase the accuracy of the model recognition. Specifically, we created a fresh backbone network for YOLOv5 that is based on the Swin transformer. Additionally, we offered a better way to combine features from different scales, and by using various detection levels, we improved the confidence loss function.

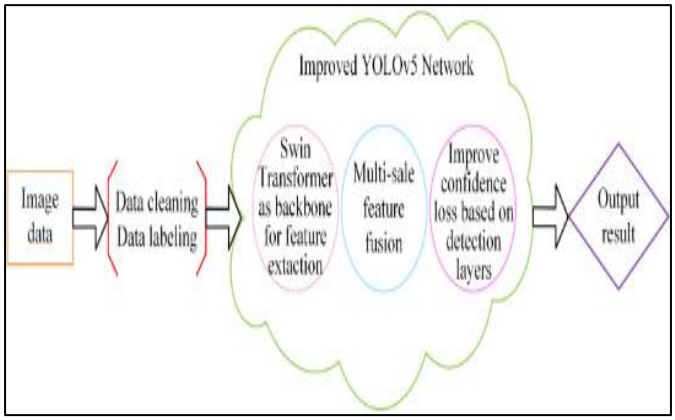

Fig 3. The improved YOLOv5 is used for underwater target detection.

2. Backbone Network Based on Swin Transformer

Images taken underwater during tracking are affected by the fact that water doesn't let all light through. This hides the targets that have been found, making it hard for the monitor to tell them apart. So, during the detecting process, the features of the targets that are being looked for should stand out more than the features that are in the background [11]. Paying attention to yourself is a good approach. Change Changer replaces the recurrent layers that are usually used in encoder-decoder designs with multi-headed self-attention, which works well in the field of natural language processing [12]. Transformer was first used in the picture world by Changer of Vision. TPH-YOLOv5 The forecast header now has Transformer encoder blocks instead of some of the convolution blocks or CSP bottleneck blocks that were in the first version of YOLOv5. These blocks helped us find targets in scenes where UAVs were used. When Transformer is used directly in the area of computer vision, there are two problems. (1) In both areas, the feature sizes used are different. It doesn't change when you do natural language processing. The feature size changes a lot in computer vision, though. Natural language processing doesn't need as high of a resolution as computer vision does. Also, using Transformer directly in computer vision can be hard on the computer because it uses the square of the picture resolution. Also, underwater devices don't have a lot of computing power, so Transformer can't be used to find targets underwater.

Swin Transformer is a successful way to use self-attention in computer vision, and it's better than earlier work in the following ways [13]: We are going to talk about three things: (1) a method used by CNN to build a hierarchical Transformer; (2) the idea of locality to do self-attention calculations within the window region without overlap; and (3) a shifted window partitioning method to make the window-based self-attention module connection work. Based on the work above, the processing difficulty goes up linearly with the size of the original picture. As the level goes up, picture blocks are slowly put together to make a Transformer that can be used for anything as a visual network that holds everything together.

Figure 4 shows how the front end of the network that was made on the Swin Transformer is put together. Patch division and linear embedding are the two parts that make up patch

embedding. The feature-map module is split into small pieces that don't touch each other. The input features are then put into any number of dimensions using linear embedding. This block is made up of W-MSA (window multi-head self-attention) and SW-MSA (shifted-window multi-head self-attention). It's easier to do the math with the W-MSA because it splits the feature map, and data can move between screens with the SW-MSA. Patch joining is used to reduce the size of the raw feature map. The first step is to give the patch embedding module the original c h a w feature map. The feature map is then broken up into small pieces that don't touch each other to make a 96 × (h/4) × (w/4) feature map. First, this map is put into two stacked Swin Transformer block modules. This makes a new 96 × (h/4) × (w/4) feature map. Next, feature maps 3, 4, and 5 are made with the help of three patch merging levels and the Swin Transformer blocks. Five feature maps are fed into the FPN (feature pyramid networks) section's neck. There are three of them: 3, 4, and 5. This picture (Figure 5) shows how YOLOv5 is put together when Swin Transformer is the main network.

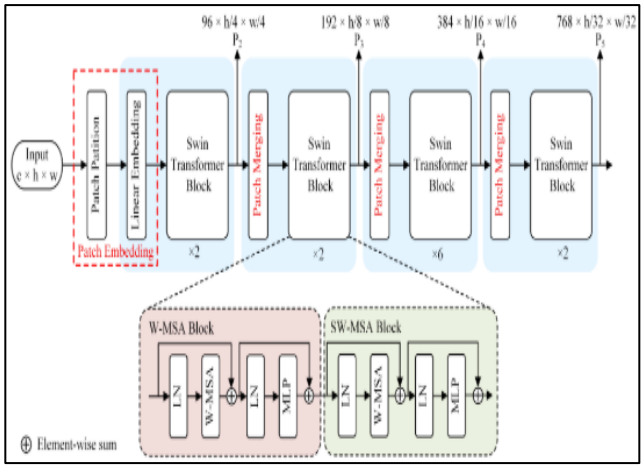

Fig 4. The Swin Transformer architecture [14].

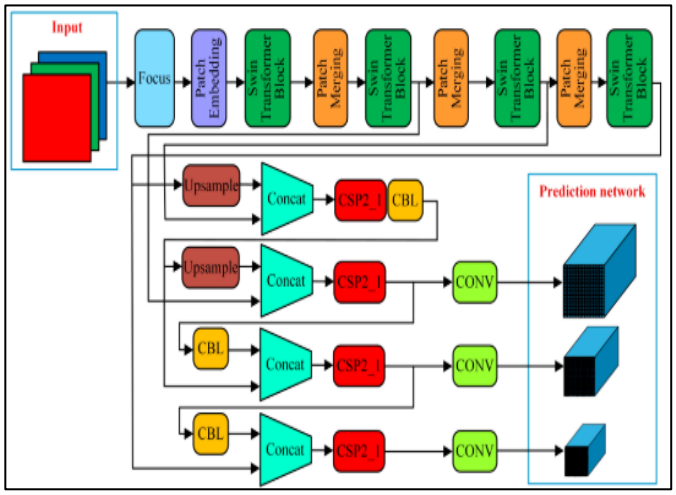

Fig 5. The structure of YOLOv5 using Swin Transformer as the backbone network.

YOLOv8 (2021)

YOLOv8 is the most recent version in the YOLO family, and it includes a few useful advancements in occlusion handling, learning multi-scale features, and processing speed. YOLOv8 has been optimized to detect smaller items, which is especially helpful in detecting trash under water when you are detecting microplastics and smaller detritus, as these types of trash can be particularly hard to detect. YOLOv8 can accommodate processing the changes in visual representation related to dynamic and messy under water environments in which many different kinds of trash and debris are within the same frame. Because it can work with bigger datasets and is faster, YOLOv8 is without a doubt one of the best tools for real-time underwater trash recognition apps.

1. YOLOv8 Network Architecture

YOLO is a popular real-time object detection system, initially developed by Joseph Redmon and others in 2016. YOLO network architecture is based on classifying and detecting objects in a single pass, as compared to many earlier frameworks that revered object localization by classifying images in multiple passes. The development of YOLO was a disruptive innovation for the computer vision space, and it is exceptionally fast and efficient at detection. In January 2023, Ultralytics launched YOLOv8, which marked a new generation of YOLO technology. YOLOv8 comes in different forms that can be used for different visual jobs. YOLOv8 has a backbone network that is similar to YOLOv5, and it also has a new module called C2f that recalls traits in a context to improve recognition. Figure 1 shows how YOLOv8 is put together. From picture processing to recognition output, the flow is shown in the figure. The first step is visual input data, the visual input will be pre-processed, using model-selective augmentation and resizing methods. When the model receives the image for feature detection, the pre-processed image will have gone through several pre-processing techniques. This is the main job of the machine: feature extraction. After the picture has been pre-processed, it is sent to the backbone network.

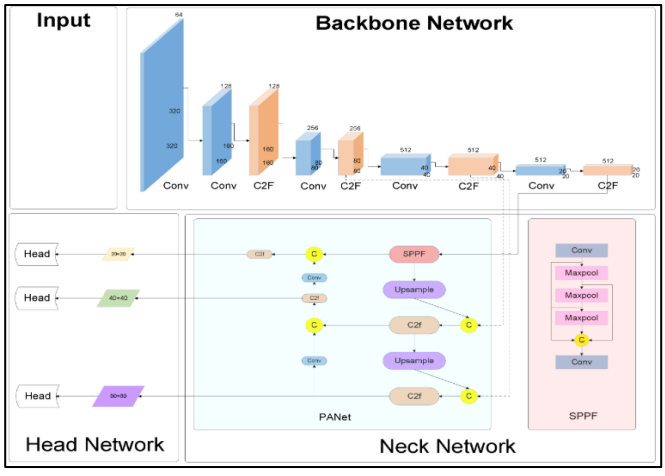

Fig 6: **YOLOv8 Network Architecture** [15]

The neck of the network is where the central feature extraction unit links the extracted features in a single extended channel. The architecture of the network with the neck configuration is designed to detect features at three different scales (small - 20 x

20, medium - 40 x 40, large-80 x 80), optimizing the detail of the required features of the differing sizes of the object. The last aspect of this cycle is the use of this multi scale outputs which are determined by the aspect of the features examined in the head of the network, which is an important step of the detection result, in deciding how to bring the promissing features together as a showcase of the detection potential of the YOLOv8 network.

Advantages of YOLO for Underwater Trash Detection

- **Speed**: One great thing about YOLO is that it lets you move quickly. Because the whole picture is detected in a single pass, data can be sent almost instantly in real time, which is very important for real-time tasks like keeping an eye on underwater trash in marine settings that are always changing.
- **Accuracy**: Mean Average Precision (mAP) is a measure of how well the model is finding and categorizing things. YOLO gives you a high MAP. YOLO has a deep architecture and offers multi-scale potential, allowing it to detect various object types accurately, from large floating debris down to small items like bottles and fishing gear.
- **Single Inference**: YOLO applies detection by thinking about the image in one inference. The methodology doesn't inspects each region separately which saves a lot of time. This aspect of YOLO is crucial for operational requirements for underwater detection which usually has to done in relatively tight timescales.
- **Multi-object Detection**: YOLO can detect multiple objects simultaneously in a single image. This ability is essential in underwater environments where various types of debris may be present in the same frame, enabling the system to identify and classify multiple pieces of trash in real-time.

Challenges of YOLO in Underwater Trash Detection

- **Small Object Detection**: One big problem for YOLO is that it has to be able to find small things. Small pieces of trash like microplastics can be hard to find underwater, especially since underwater pictures don't have a lot of detail. While YOLOv8 has made strides in improving small object detection, underwater trash like microplastics remains a challenge due to pixel limitations and low contrast against the background [16].
- **Localization Errors**: YOLO sometimes struggles with the accurate localization of objects, especially when they are overlapping or partially occluded. This issue becomes more apparent in underwater settings, where light refraction and murkiness can obscure the true location and shape of debris. Mislocalization can lead to errors in detection and classification, affecting the effectiveness of real-time monitoring systems [17].

FASTER R-CNN

2.2. Network Structure of the Faster RCNN

In general, the Faster RCNN is made up of three main parts: getting feature information from the input picture, drawing bounding boxes, classifier classification, and the regressor adjusting the position of objects. Figure 2 shows the main parts of the faster RCNN. A picture can be used to teach a convolutional neural network how to get knowledge about features. The neural feature information is then sent to the RPN (area Proposal Network), which makes area proposals. The regression layer's job is to guess the area plan parameters that go with the bounding box's reference points. Find out if the thing inside the box is an object or background. This is the classification layer's main job. The neural feature map can be used to connect the RPN-suggested areas to the ones that are already there to make a ROI. With the sharing process of the ROI, the mapped ROI areas are then split into blocks of the same size. Last but not least, the highest sharing action changes the size of the boxes around each area. Finally, data about the edges of each area must be sent to the next level of the network, which is the fully linked layer. When it gets to this layer, the softmax function can show the label classification score and where the updated bounding box is [18].

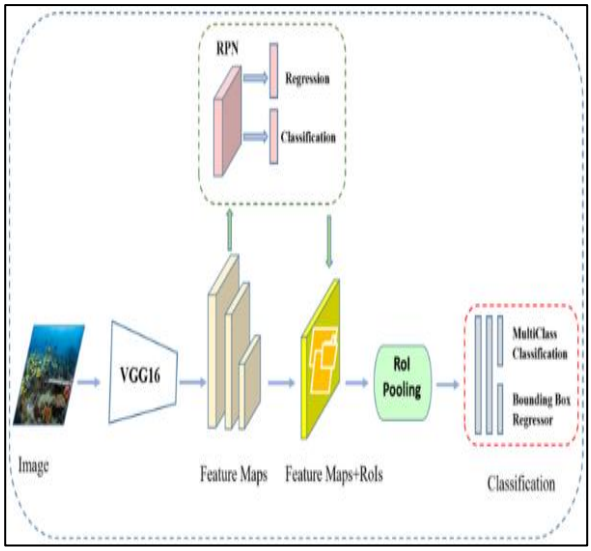

Fig 7. Network structure diagram of the Faster RCNN.

2.3. Loss Function of the Faster RCNN

For the regional network, bounding box regression loss is part of the Faster RCNN's loss function. This loss is used for classification.The other part is classification loss, which includes loss of bounding box position adjustment at detection. Equation can be used to describe the Faster RCNN's loss function [19].

The loss function L is given by:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}\,(p_i, p_i^*) + \frac{1}{N_{reg}}\sum_i p_i^*\, L_{reg}(t_i, t_i^*)$$

Where:

- i is a list of the anchor numbers in each small set of data.
- p_i is the chance that the ties to things will work as predicted.

- One zero equals an object that is not true, and one one equals an object that is true. That is what p_i^* means.
- λ is a measure of weight.
- N_cls is a measure for classification loss. One of the regression loss parameters is N_reg [20].

$L_{cls}(p_i, p_i^*)$ is the logarithmic loss between the object and non-object, calculated as $L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)]$

R is a function named smoothL1(x), as shown in Equation:

$$smoothL1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Where:

- $L_{reg}(t_i, t_i^*)$ is the regression loss inside the object detection, represented as $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$,
- t_i refers to the predicted coordinates of the object.
- t_i^* refers to the true coordinates of the detection object.
- $x = t_i - t_i^*$, where x is the difference between predicted and actual coordinates.

This equation represents a loss function for training an object detection model, integrating classification and regression components, where the regression loss uses the smooth L1 loss function for more robust prediction of object coordinates.

3. Image Enhancement Techniques

1. Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE is widely used for underwater image enhancement due to its ability to improve local contrast in regions where global histogram equalization would fail to produce satisfactory results. This technique divides the image into small tiles (or blocks), processes each tile separately, and then combines them to create a uniform enhancement. CLAHE is particularly effective in underwater imaging as it can prevent over-enhancement in uniformly lit areas and suppress noise amplification.

How CLAHE Improves Image Quality for Object Detection:

- **Local Contrast Enhancement**: CLAHE makes underwater items easier to see by raising the contrast in small parts of a picture without changing the overall image. This localized approach ensures that small or faint features are enhanced without over processing the entire image.
- **Noise Suppression**: CLAHE reduces the likelihood of noise amplification. This is essential in murky, low visibility, underwater photographs in which global enhancement techniques might also increase noise, leading to poorer detection.
- **Feature Enhancement:** With this trick, we can improve the visibility of small underwater objects, such as debris or marine life, making it easier for object detection models like YOLO or Faster R-CNN to classify them correctly.

2. Denoising and Contrast Enhancement

Underwater images can be distorted due to noise, blur, and low contrast. To mitigate these problems, there are preprocessing methods available that aim to improve image quality for object detection, such as, histogram equalization, linear contrast adjustment, and denoising methods.

Role of Preprocessing Methods:

- **Histogram Equalization**: This approach changes the intensity values to enhance the contrast of an image. Unfortunately, it can add unwanted noise, particularly in low-light environments underwater. Commonly modified techniques of AHE and CLAHE offer improved localized enhancement.
- **Linear Contrast Adjustment**: This technique spreads the pixel values out as much as possible along the intensity range, which increases contrast; however, it may not work well when the quality of the image is affected by poor lighting conditions or color distortion caused by the light refraction and scattering encountered in the underwater environment.
- **Denoising**: Denoising methods, such as Gaussian blurring, median filtering, and wavelet-based methods, serve to reduce are enhancing clarity in noisy underwater images while preserving details. A more complex denoise method, Non-Local Means Denoising, improves the clarity of images while preserving edges which is important in properly recognizing objects.

3. Advanced Denoising Techniques for Underwater Imaging

To rectify the problems of noise and blur in underwater situations, advanced denoising techniques are combined with contrast enhancement approaches to enable a more robust preprocessing of underwater images. For example, Bilateral Filtering, Wavelet Transform, and Non-Local Means have all been implemented specifically to reduce noise and preserve fine details (like edges) which is important for object detection.

- **Bilateral Filtering**: This method smooths images and preserves edges, thus making it useful for improving the visibility underwater without blur the item of interest. The bilateral filter considers both the spatial and intensity proximity of the pixels to reduce noise without losing the sharpness of the photo.
- **Wavelet Transform Denoising**: The wavelet transform is a particularly strong denoising technique for underwater imagery because it breaks down the image into different frequency bands. The process of denoising can focus on the high-frequency parts that are noise while getting the low-frequency parts that are important features, typically without missing or ruining most of the low-frequency parts.
- **Non-Local Means Denoising**: Non-Local Means (NLM) is a complex image-denoising algorithm that compares each patch to find similar ones in the picture. It does this while keeping more of the signal's data than other denoising algorithms. NLM is particularly effective in removing noise while maintaining the finer textures and details that underwater environments offer.

4. Challenges in Underwater Trash Detection

1. Low Visibility and Motion Blur

Low visibility underwater is one of the biggest challenges for detecting trash. Water turbidity, variable lighting situations, and suspended particles can all lead to a decrease in visibility underwater making things more cloudy and object blur or motion blur may have a significant effect on the data quality leading to much lower-quality data, and ultimately poorer accuracy using detection algorithms. Visibilities may be converted from bad to worse with the motion blur caused by water currents. If the water is already disturbed, the natural disturbance combined with the motion blur makes it difficult for detection systems to detect things when it can't see them distictly. Therefore, identifying and classifying the underwater trash becomes convoluted as many debris objects may appear warped, half buried, or-all together out of focus from the image or video data.

2. Dataset Limitations

Another big problem with finding trash underwater is that there aren't enough big datasets with labels that can be used to train deep learning models. Underwater trash detection suffers from the unavailability of datasets with the quality of datasets used for many other computer vision tasks. Existing underwater trash detection datasets often lack quality, comprehensiveness, or a diversity of marine debris. Other datasets also lack sufficient annotation for the effective training of models to identify the various types of trash, which can range in size from large objects like plastic bottles and fishing gear to smaller items like microplastics. Unfortunately, not having enough complete and varied datasets makes models too good at detecting certain types of debris and makes them less useful in other situations. This has a big effect on how deep learning models designed for finding trash underwater are used in all places and conditions.

3. Real-time Processing and Resource Constraints

A fundamental challenge for using an underwater trash detection system is the capability of processing in real-time with resource constraints. In situations where processing and power are restricted, underwater vehicles like remotely operated vehicles (ROVs) or autonomous underwater vehicles (AUVs) are often used. The vehicles actually run off of onboard systems, which can only do so much and, in some cases, put restrictions on the onboard data. The detection algorithms should be optimized to allow for rapid processing, since there is little leeway or tolerance for performance with processing. A detection algorithm could run the risk of losing detection opportunity, if not swift enough for processing. Additionally, the powered vehicle imposes further constraints on processing computational expense, limiting the use of elaborate models. Light-weight and efficient algorithms for detection models are essential if they are to run in real-time with minimal latency. The algorithms will need to provide instantaneous on-board results within the power constraints. Without execution of efficient algorithms, the task of continuous and exhaustive monitoring of trash underwater becomes impossible especially in long missions or remote areas with limited opportunity for recharging or transmitting data.

2. FUTURE SCOPE

The future of underwater trash detection using deep learning algorithms lies in overcoming the existing challenges related to environmental variability, dataset limitations, real-time processing, and computational constraints. One major avenue for future research is the integration of multimodal data, including sonar, infrared, and acoustic signals, to enhance the robustness of deep learning models in diverse underwater conditions. This could mitigate issues caused by water turbidity, varying light conditions, and refraction that hinder the performance of visual models. Further advancements in model architectures, such as hybrid models combining CNNs, transformers, and attention mechanisms, are also critical for improving accuracy, particularly in detecting small and overlapping objects like microplastics. These advancements will enable models to perform better in the presence of occlusions and rare debris, leading to better classification and detection of mocking trash in the water. In addition to this, larger and more extensive annotated datasets are needed, especially for smaller debris types. Data augmentation techniques, encompassing synthetic data generation, and adversarial learning efforts, can aid in facilitating a lack of data, which can improve models' capability of generalizing any environment. Another big problem in underwater trash recognition is that there aren't many real-time uses. This is especially true for autonomous underwater vehicles (AUVs) and remotely controlled vehicles (ROVs). In the future, work can be focused on making lightweight versions of deep learning models and improving the model optimization processes so that they use less power and computer processing while still being able to identify things within a good range. This could lead to better continued incorporation of edge computing or cloud processing to inform real-time detection and analysis. By addressing these challenges, and seeking these new avenues, deep-learning algorithms could continue to build on the current iterations of underwater trash detection systems and contribute to improved environmental monitoring or marine conservation initiatives.

3. CONCLUSION

In summary, advancements in deep learning algorithms have greatly improved the quality of underwater trash detection methods, primarily offering enhancements to existing practices like sonar or manual inspection. Models like YOLO, Faster R-CNN and their more recent versions, including YOLOv8 have achieved promising performance with real-time detection and classification tasks and have proven effective in complex underwater environments. These algorithms provide very robust and accurate detection of most marine debris, including both large floating debris and smaller microplastics, when trained using large, varied datasets. But there are still problems that need to be fixed before deep learning can be used to successfully find trash underwater using pictures of the ocean. processing is going to be a challenge in terms of computational Environmental issues, including clarity of the water, variable light conditions, and surface interference all can affect the images taken underwater, resulting in detection errors. In addition, underwater researchers face the problem of acquiring a large annotated dataset of high quality, especially for small

debris, limiting the ability of any trained model to generalise across various underwater environments. It will be hard to do real-time processing because of limited computer power, especially for unmanned underwater vehicles (AUVs) or ROVs that are controlled from afar. Despite these sticking points, the future of underwater trash detection using deep learning is still promising. Future developments are expected to focus on fusing multimodal sensor data, optimizing model architectures, improving augmentations for small object detection and efficiency; developing lightweight, efficient models that can be used to enable real-time detection on constrained platforms or limited computational resources. By fixing these problems, deep learning algorithms should always be able to make underwater trash tracking systems more accurate, scalable, and efficient. This would give more accurate signs of protecting the oceans and reducing pollution.

4. REFERENCES

[1] J. R. Jambeck *et al.*, "Plastic waste inputs from land into the ocean," *Science*, vol. 347, no. 6223, pp. 768–771, Feb. 2015, doi: 10.1126/science.1260352.

[2] A. Cózar *et al.*, "Plastic debris in the open ocean," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 111, no. 28, pp. 10239–10244, Jul. 2014, doi: 10.1073/pnas.1314705111.

[3] M. Mathangi, "Detection of Underwater Trash Objects using Deep Learning Algorithms," 2023.

[4] S. Oron, A. Sadekov, T. Katz, and B. Goodman-Tchernov, "Benthic foraminifera geochemistry as a monitoring tool for heavy metal and phosphorus pollution — A post fish-farm removal case study," *Marine Pollution Bulletin*, vol. 168, p. 112443, Jul. 2021, doi: 10.1016/j.marpolbul.2021.112443.

[5] B. Wang, L. Hua, H. Mei, Y. Kang, and N. Zhao, "Monitoring marine pollution for carbon neutrality through a deep learning method with multi-source data fusion," *Front. Ecol. Evol.*, vol. 11, p. 1257542, Sep. 2023, doi: 10.3389/fevo.2023.1257542.

[6] W. Ge, J. Sun, Y. Xu, and H. Zheng, "Real-time Object Detection Algorithm for Underwater Robots," in *2021 China Automation Congress (CAC)*, Beijing, China: IEEE, Oct. 2021, pp. 7703–7707. doi: 10.1109/CAC53003.2021.9727507.

[7] A. Akib *et al.*, "Unmanned Floating Waste Collecting Robot," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Kochi, India: IEEE, Oct. 2019, pp. 2645–2650. doi: 10.1109/TENCON.2019.8929537.

[8] Pragati Baheti, "The Essential Guide to Neural Network Architectures." 2021. [Online]. Available: https://www.v7labs.com/blog/neural-network-architectures-guide

[9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, *arXiv*. doi: 10.48550/ARXIV.1804.02767.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, *arXiv*. doi: 10.48550/ARXIV.2004.10934.

[11] A. Vaswani *et al.*, "Attention is All you Need".

[12] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Jun. 03, 2021, *arXiv*: arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929.

[13] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," 2021, *arXiv*. doi: 10.48550/ARXIV.2103.14030.

[14] F. Lei, F. Tang, and S. Li, "Underwater Target Detection Algorithm Based on Improved YOLOv5," *JMSE*, vol. 10, no. 3, p. 310, Feb. 2022, doi: 10.3390/jmse10030310.

[15] J. Zhu, T. Hu, L. Zheng, N. Zhou, H. Ge, and Z. Hong, "YOLOv8-C2f-Faster-EMA: An Improved Underwater Trash Detection Model Based on YOLOv8," *Sensors*, vol. 24, no. 8, p. 2483, Apr. 2024, doi: 10.3390/s24082483.

[16] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, *arXiv*. doi: 10.48550/ARXIV.1804.02767.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, *arXiv*. doi: 10.48550/ARXIV.2004.10934.

[18] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," 2016, *arXiv*. doi: 10.48550/ARXIV.1611.01144.

[19] Q. Xu, X. Zhang, R. Cheng, Y. Song, and N. Wang, "Occlusion Problem-Oriented Adversarial Faster-RCNN Scheme," *IEEE Access*, vol. 7, pp. 170362–170373, 2019, doi: 10.1109/ACCESS.2019.2955685.

[20] G. Hahn, S. M. Lutz, N. Laha, and C. Lange, "A framework to efficiently smooth L_1 penalties for linear regression," Sep. 19, 2020, *Bioinformatics*. doi: 10.1101/2020.09.17.301788.

AUTHORS:

Jyoti Sandur a Postgraduate in Electronics Engineering – VLSI & Embedded Systems from JSPM Narhe Technical Campus, SPPU, Pune. Currently, pursuing PhD on Underwater Trash Detection using Deep Learning Algorithms from MITWPU, Pune and working as an Assistant Professor with Bharati Vidyapeeth Deemed to be University College of Engineering, Pune with a teaching experience of 9 years and industrial experience of 2.5 years. Her area of interests includes soft computing, Artificial Neural Network and Biomedical signal processing.

Corresponding Author E-mail: jyoti.sandur@gmail.com

Dr. Alka Barhatte a Postgraduate in Electronics Engineering – VLSI Design from Bharti Vidyapeeth Deemed University, Pune and doctorate in Electronics and Telecommunication by SPPU. I am currently working as an Assistant Professor at the School of ECE, MITWPU with teaching experience of 16 years and research experience of 8 years. Her area of interests includes soft computing, Artificial Neural Network and Biomedical signal processing.

E-mail: alka.barhatte@mitpune.edu.in Author