“UNDERWATER”

INTELLECTUAL OBJECT DETECTION SYSTEM FOR AUTONOMOUS UNDERWATER VEHICLES

CONTENTS

ABSTRACT

The use of intelligent underwater vehicles for ocean exploration has been highly effective in areas like marine life exploration, fisheries, ecological monitoring, and military applications. Integrating computer vision algorithms has greatly enhanced these vehicles' capabilities. However, underwater image detection faces unique challenges such as poor image quality, densely-packed targets that are difficult to discern, limited training data, and constrained computational power. There is a **need for research** of innovative approaches specifically tailored to address these challenges and enhance the precision and efficiency of underwater object detection.

The purpose of this study is to design an intellectual system for underwater object detection and classification, which is both deployment-efficient and achieves high accuracy.

The tasks are as follows:

1) analyze the challenges and existing approaches, find out its limitations;
2) create a preprocessing module to effectively process underwater images;
3) design a neural network topology to resolve unique challenges present in the realm of underwater object detection;
4) train the network on data, obtained from intelligent underwater vehicle camera;
5) analyze the results and compare it to existing results;
6) build software for the proposed approach.

Methodology used in this paper: we proposed a deployment-efficient intelligent system for underwater object detection handled autonomously on such a device, which achieves superior performance by integrating preprocessing module, custom feature extractor with transformer block, neck module with attention layers and an improved YOLO-based object detection head module, which integrates extra prediction head for small objects, based on higher-resolution feature maps.

Keywords: underwater object detection; classification; hybrid neural networks, transformers; attention; deep learning

1. RELATED WORK

In the realm of object detection, while general object detection algorithms have demonstrated proficiency on generic datasets, their application to underwater scenes poses distinct challenges. These challenges stem from the inherent complexities of underwater environments, including poor image quality, color distortion, light interference, and the prevalence of small, densely-packed targets. Consequently, the task of UOD necessitates a nuanced approach, typically bifurcated into image pre-processing and object detection subtasks.

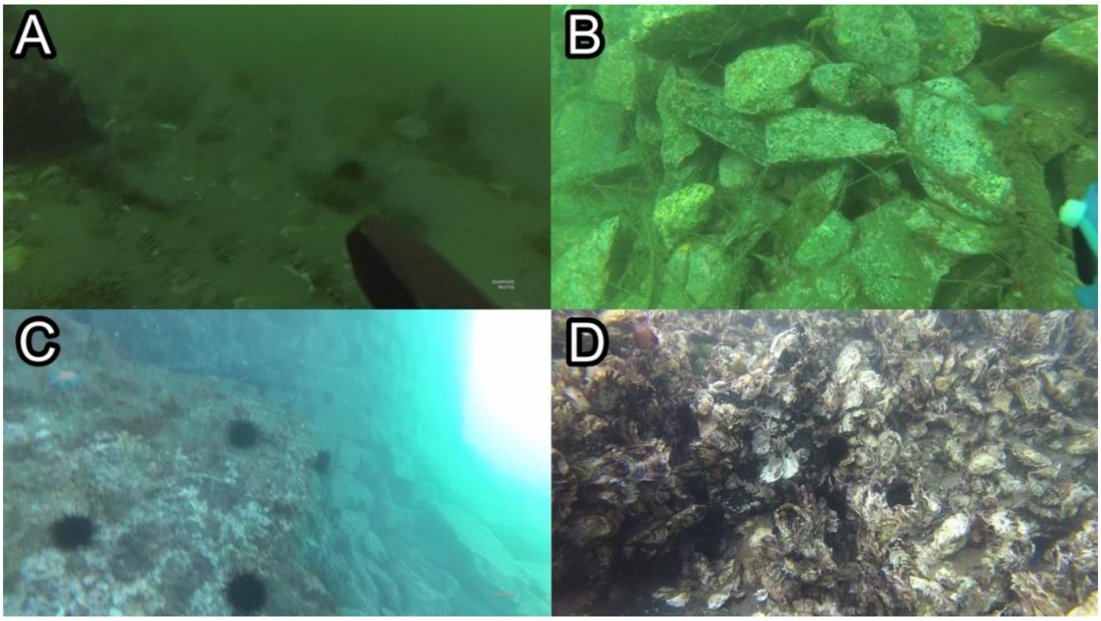


Fig. 1. Samples from UTDAC2020 dataset demonstrating unique challenges in underwater object detection tasks, such as a) low contrast, b) presence of small and densely located objects, c) lighting issues, d) target occlusion and mimicry.

1.1 Image preprocessing

Underwater visibility is profoundly impacted by the presence of water molecules and suspended particles, which introduce distortion to light and result in the absorption of different colors at varying wavelengths. To address this challenge, recent advancements have been made to enhance underwater neural networks using innovative techniques. The researchers in [1] leveraged the absorption differences in various color channel wavelengths to estimate the transmission plot through the scene depth. The underwater neural network was then fortified through the application of graph tangent theory, demonstrating a promising approach to tackle the adverse effects of underwater visibility. In a complementary effort, Hu et al. [2] conducted a thorough analysis of the imaging principles underlying underwater images and outlined the factors contributing to their diminished quality. They succinctly categorized existing methods and delved into underwater video enhancement technologies. Notably, underwater imagery is susceptible to strong absorption, scattering, color distortion, and noise stemming from artificial light sources, resulting in image blur, haziness, and a discernible bluish or greenish tone. To mitigate these issues, a significant contribution was made in [3], where two distinct methods for underwater image dehazing and color restoration were proposed, offering valuable insights into combating the challenges associated with underwater image quality degradation. Hu et al. [4] introduced an improved approach for correcting transmittance based on the underwater polarization imaging model, mitigating inaccuracies in object irradiance caused by polarization effects. He et al. [5] presented the dark channel prior algorithm, which estimates light transmission maps to remove fog-induced blur by leveraging the tendency of the dark channel intensity to approach zero in clear images. Fu et al. [6] proposed a Retinex-model-based method for image decomposition, separating images into illumination and reflectance components to preserve object detail, further introducing a novel shrinkage factor to aid in component estimation. Zhang et al. [7] devised a technique involving color correction and gamma correction within the HSV color space and Retinex model, respectively, to restore the true appearance of underwater images. Lastly, Liu et al. [8] introduced the twin adversarial contrastive learning algorithm, which employs both self-supervised and unsupervised approaches to process underwater datasets.

In their comprehensive framework for underwater image enhancement detailed in [9], the researchers presented a sophisticated combination of techniques aimed at addressing the multifaceted challenges inherent in underwater imagery. The proposed algorithm encompasses an improved image fusion and enhancement strategy including image edge feature sharpening and dark detail enhancement through homomorphism filtering conducted within the CIELab color space. To counter color deviation and elevate color saturation, the multi-scale retinal with color restoration (MSRCR) algorithm is employed within the RGB color space. Additionally, the contrast-limited adaptive histogram equalization (CLAHE) algorithm is enlisted to effectively defog the images and enhance overall contrast, ultimately yielding a final enhanced image that represents a significant advancement in underwater image quality. Hong et al. [10] introduced the Water Quality Transfer (WQT) augmentation method, augmenting domain diversity and enhancing the performance of domain generalization in UOD. Lin et al. [11] presented an image enhancement algorithm based on candidate frame fusion, refining the detection of underwater targets. Sun et al. [12] employed transfer learning techniques to achieve exceptional results in identifying objects in low-quality underwater videos, boasting an impressive average classification accuracy of 99.68% for 23 fish species.

1.2 Object detection and classification

In the domain of deep-learning-based object detection models, two primary methodologies have emerged: anchor-based algorithms and anchor-free algorithms. Anchor-based approaches, such as Faster R-CNN [13], SSD [14], and RetinaNet [15], rely on predefined anchor boxes to localize objects within images. Anchor-free algorithms as YOLOX [16] and FCOS [17] only calculate the center point of the bounding box and position coordinates compared to the pre-set anchor scale and aspect ratio., simplifying the detection pipeline. Recent advancements such as attention have further enhanced the capabilities of these methods, refining object localization and classification accuracy. Notably, in the context of underwater image analysis, researchers often adopt a holistic approach, integrating both image enhancement and object detection components into their models.

The challenges posed by small and dense underwater targets necessitate advanced techniques for effective object detection. Deep Convolutional Neural

Networks (CNNs) have emerged as powerful tools for such tasks, enabling multi-layer non-linear transformations that extract intricate features and represent them at higher abstraction levels. Among the widely-used CNNs, the YOLO (You Only Look Once) series stands out for its efficiency in object detection tasks [18].

1.3 Underwater object detection frameworks

Researchers have leveraged YOLO-based architectures for various underwater applications, showcasing the versatility of these networks. Sung et al. [19] proposed a YOLO-based CNN for real-time video image analysis, achieving an impressive 93% classification accuracy in detecting fish. Pedersen et al. [20] adopted YOLOv2 and YOLOv3 for marine-animal detection, while Zhang et al. [21] introduced a lightweight UOD framework based on YOLOv4 and multi-scale attentional feature fusion. Liang Chen et al. [22] contributed to the field with a lightweight underwater target detection algorithm based on dynamic sampling transformer and knowledge distillation optimization. In a novel approach [23], Liu et al. introduced TC-YOLO, a new UOD framework. This framework combines the CLAHE preprocessing algorithm, a modified YOLOv5s architecture, and optimal transport label assignment with attention mechanisms in the backbone and neck of the network, showcasing a comprehensive strategy to address the intricacies of UOD. Shen et al. [24] proposed the multi-dimensional, multi-functional, and multi-level attention module (mDFLAM), enhancing the robustness and generalization capabilities of YOLO on underwater images. Xu et al. [25] devised the scale-aware feature pyramid architecture named SA-FPN, optimizing feature extraction and enhancing marine object detection performance. Pan et al. [26] introduced a modified method based on multi-scale ResNet, improving efficiency by accurately detecting objects of various sizes, particularly small ones. Muksit et al. [27] enhanced the original YOLOv3, addressing issues of misdetection of tiny fish by adjusting upsampling step sizes and incorporating Spatial Pyramid Pooling. Long Chen et al. [28] proposed the SWIPENet algorithm, leveraging sample re-weighting to reduce interference from noisy samples. Lingyu Chen et al. [29] modified the YOLOv4 structure by integrating deconvolution modules and depthwise separable convolution, enhancing the network's capabilities. Wang et al. augmented YOLOv7 [30] with an image enhancement module and implemented Focal EIOU as a new bounding box regression loss, mitigating performance degradation caused by mutual occlusion and overlapping of underwater objects. Lastly, Minghua Zhang et al. in [31] proposed to replace the original backbone of YOLOv8 with FasterNet network, which is optimized for low latency. However, the proposed algorithm lacks underwater image enhancement network, which results into missing certain targets in environments with poor visibility and high degree of target overlap.

1.4 The limitations of existing underwater object detection systems

Original YOLOv8 network is considered a state-of-the-art (SOTA) model for object detection, and provides several major improvements compared to earlier versions. In backbone, C3 structure has been replaced with lighter C2f to facilitate more extensive gradient flow. Head part of YOLOv8 architecture has been modified by decoupling heads for regression and classification tasks, also replacing anchor-based design in favor of anchor-free. The loss function of YOLOv8 features distribution focal loss term and the task-aligned assigner label matching strategy. YOLOv8 is available in five variants, which differ in parameters count. Starting for the tiniest, the versions are YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x.

However, although YOLOv8 provides pre-made slimmed versions like YOLOv8s, it still has drawbacks like high computational complexity and high network transmission volume, which results in slower detection speed and increased hardware requirements, making it more difficult to operate on edge devices. To solve these problems, a more hardware-aware approach is required.

Regarding the systems optimized specifically underwater object detection, several drawbacks persist within current methodologies. Often, researchers concentrate their efforts on isolated aspects of the detection pipeline, such as refining image preprocessing techniques or enhancing object detection algorithms. Consequently, there is a dearth of comprehensive approaches that synergize the strengths of both domains, potentially limiting the overall effectiveness of underwater detection systems. Moreover, many proposed algorithms overlook the practical constraints imposed by the hardware limitations of underwater vehicles. These algorithms frequently fail to account for the necessity of running on resource-constrained hardware and rely heavily on pre-made, smaller versions of YOLO. This oversight hampers the scalability and applicability of these algorithms in real-world underwater environments, highlighting the need for more holistic and hardware-aware approaches to underwater object detection.

2. THEORETICAL RESEARCH

2.1 Problem statement

Given a dataset comprising images, the objective is to develop an efficient object detection and classification network capable of accurately identifying objects within the images, determining their respective bounding boxes, and assigning appropriate class labels to the detected objects.

Let $(X_1, Y_1), (X_2, Y_2), \dots, (X_i, Y_i)$ represent the training samples, where $X_i \in \mathbb{R}^{n\times n}$ denotes the $i-$th RGB image matrix with dimensions $n \times n \times 3$, and Y_i represents the ground truth annotations for the corresponding image, consisting bounding box coordinates and class labels.

The primary objective is to develop a neural network architecture capable of accurately predicting bounding box coordinates and class probabilities for objects within the input images. This involves training the network to learn optimal weight coefficients that minimize a predefined loss function.

Let $f_\theta: \mathbb{R}^{n\times n} \to \mathbb{R}^{S\times S\times(B\times 5+C)}$ represent the YOLO network parameterized by weights θ, where S is the grid size, B is the number of bounding boxes predicted per grid cell, and C is the total number of object classes. The output of the network is a tensor of dimensions $S \times S \times (B \times 5 + C)$ representing the predicted bounding box coordinates and class probabilities for each grid cell.

Generally, loss function for object detection and classification tasks can be defined as:

$$\mathcal{L} = \lambda_{\text{coord}} \cdot \mathcal{L}_{\text{coord}} + \lambda_{\text{conf}} \cdot \mathcal{L}_{\text{conf}} + \lambda_{\text{class}} \cdot \mathcal{L}_{\text{class}}$$

where $\mathcal{L}_{\text{coord}}$, $\mathcal{L}_{\text{conf}}$ and $\mathcal{L}_{\text{class}}$ are the localization, confidence and classification losses in that order, and λ_{coord}, λ_{conf}, λ_{class} are the coefficients to balance the influence of each component in general loss function.

In modern YOLOv8-based detectors, CIoU [32] is used as the box loss, binary cross entropy is used for multi-label classification as the classification loss and distribution focal loss [33] is used as the third term in general loss function. Formally, this loss function can be defined as follows:

$$
\begin{aligned}
\mathcal{L} = \frac{\lambda_{\text{box}}}{N_{\text{pos}}} \sum_{x,y} \mathbf{1}_{c_{x,y}^{*}} \left[1 - q_{x,y} + \frac{\left\| b_{x,y} - \hat{b}_{x,y} \right\|_2^2}{\rho^2} + \alpha_{x,y} \nu_{x,y} \right] \\
+ \frac{\lambda_{\text{cls}}}{N_{\text{pos}}} \sum_{x,y} \sum_{c \in \text{ classes}} y_c \log(\hat{y}_c) + (1 - y_c) \log(1 - \hat{y}_c) \\
+ \frac{\lambda_{\text{dfl}}}{N_{\text{pos}}} \sum_{x,y} \mathbf{1}_{c_{x,y}^{*}} \left[-\left(q_{(x,y)+1} - q_{x,y} \right) \log\left(\hat{q}_{\,x,y} \right) \right. \\
\left. + \left(q_{x,y} - q_{(x,y)-1} \right) \log\left(\hat{q}_{\,(x,y)+1} \right) \right]
\end{aligned}
$$

where:

$$
\begin{gathered}
q_{x,y} = IoU_{x,y} = \frac{\hat{\beta}_{x,y} \cap \beta_{x,y}}{\hat{\beta}_{x,y} \cup \beta_{x,y}} \\
\nu_{x,y} = \frac{4}{\pi^2} \left(\arctan\left(\frac{w_{x,y}}{h_{x,y}} \right) - \arctan\left(\frac{\hat{w}_{x,y}}{\hat{h}_{x,y}} \right) \right)^2 \\
\alpha_{x,y} = \frac{\nu}{1 - q_{x,y}} \\
\hat{y}_c = \sigma(\cdot) \\
\hat{q}_{\,x,y} = \text{softmax}(\cdot)
\end{gathered}
$$

Here, N_{pos} represents the number of cells featuring an object, $\mathbb{1}_{c_{x,y}^{*}}$is an indicator function for the cells featuring an object, $\beta_{x,y}$ is the ground truth bounding box position, $b_{x,y}$ is the predicted box of the respective cell, $\hat{\beta}_{x,y}$ are the coordinates of the center point of the ground truth bounding box, y_c represents the ground truth label for class c for each individual grid cell (x, y) in the input, $q_{(x,y)+1}$ are the nearest left and right predicted boxes IoU which belong to $c_{x,y}^{*}$, $w_{x,y}$ and $h_{x,y}$ are width and height of the box, ρ is the diagonal length of the smallest enclosing box covering the predicted and ground truth boxes. The best candidate is then determined by each cell for predicting the bounding box of the object.

The optimization process involves minimizing the defined loss function $\mathcal{L}$ using Stochastic Gradient Descent (SGD) with momentum. The update rule for SGD with momentum is as follows:

$$v_{t+1} = \beta v_t + (1-\beta)\nabla_\theta \mathcal{L}(\theta_t)$$

$$\theta_{t+1} = \theta_t - \eta v_{t+1},$$

where v_t is the velocity term representing the exponentially weighted moving average of past gradients, θ_t are the network parameters at iteration t, $\nabla_\theta \mathcal{L}(\theta_t)$ is the gradient of the loss function with respect to the network parameters at iteration t, η denotes the learning rate, β represents the momentum term.

2.2 Proposed approach

To solve the unique challenges, present in underwater object detection tasks, such as visibility issues, the presence of small densely packed objects and target occlusion, we present a complex object detection system. Its architecture is based on YOLOv8 design, which consists of three parts: backbone, neck and head. In brief, the following modules have been developed to create our underwater object detection system:

1. **Preprocessing module:** to reduce the unwanted distortion effects which are present in underwater images, we have designed a preprocessing module, which decreases false negatives count, especially with smaller targets.
2. **Backbone with Transformer block:** to overcome a significant constraint of YOLOv8 in adequately capturing global and contextual information, which is a frequent issue encountered by CNNs with restricted receptive fields, our methodology entails the replacement of the terminal C2f layer within the CSPDarkNet53 backbone with a Swin Transformer block.
3. **Neck with attention module:** the attention modules were embedded in the neck part of object detection system to capture the global information for the features of different sizes after its extraction and fusion. This integration provides further improvement in object detection capabilities of the system. The attention block was carefully selected to be both effective and deployment efficient.

4. **Object detection head for small objects**: through experimenting with different feature map sizes, we have discovered that using larger feature maps helps significantly with the detection of small objects, thus we have developed an additional classification head accepting larger 160 x 160 feature map as an input, which resulted in improved small target detection.

2.2.1 Preprocessing module

We have developed an image preprocessing module to introduce a boost in underwater model recall by improving the details and restoring original colors of the underwater image, which results in a decreased number of missed detections when it comes to smaller targets. The module is based on Contrast Limited Adaptive Histogram Equalization (CLAHE) [34], as it was proven to be most effective preprocessing solution during comparison with Histogram Equalization (HE) and Adaptive Histogram Equalization (AHE).

	Algorithm	Precision	Recall	Time
1	Original	82.73%	78.96%	-
2	HE	80.8%	76.67%	2.5ms
3	AHE	78.67%	75.22%	2.7ms
4	CLAHE	82.67%	81.02%	2.5ms

Table 1. Preprocessing algorithms influence on object detection performance

CLAHE algorithm and consists of four steps:

1) Divide the image into non-overlapping tiles of a specified size.
2) Compute the histogram $\mathrm{H_i(k)}$ for each tile i.
3) Perform histogram equalization independently for each tile:

$$I_{\mathrm{eq}}(x, y) = \mathrm{L}(I(x, y), H_i)$$

where L is the function that maps the intensity values of $I(x, y)$ to the corresponding equalized values using the histogram H_i.

4) Apply contrast limiting:

$$\mathrm{I}_{\mathrm{clahe}}(x,y) = \begin{cases} I_{\mathrm{eq}}(x,y) \\ I_{\mathrm{eq}}(x,y) + \frac{I_{\mathrm{eq}}(x,y) - \mathrm{clipLimit}}{N_i} \end{cases}$$

The images from training and testing set were resized to 640 x 640 and then processed using CLAHE implementation from OpenCV library with ‘clip limit’ value manually lowered to 2 to avoid unnecessary changes in color. Average processing time for each image sample is 2.5ms, and the boost in recall comparing to an unprocessed image is 2.06% which we consider an acceptable trade-off between time and performance.

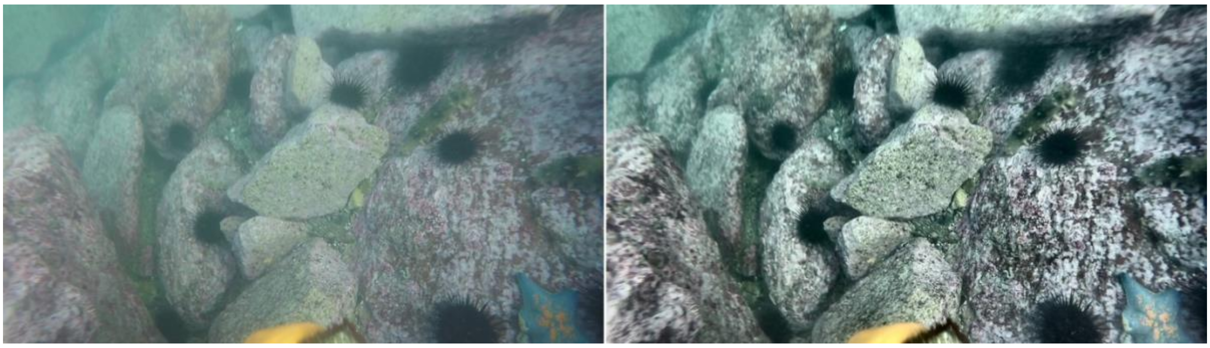


Fig. 2. Image taken from UTDAC2020 dataset before and after preprocessing

2.2.2 Modified network backbone

We have designed a modified version of CSPDarkNet53 backbone featuring a Swin Transformer block, which replaces the terminal C2f block in original CSPDarkNet53. The placement of the block is based on an assumption that operating on low-resolution feature maps (20 x 20) would reduce computational complexity and memory requirements. Integrating this block results in better capturing global and contextual information, improving the detection performance with the objects of varying sizes.

Swin Transformer uses a patch division module to split the image into non-overlapping segments and treats it as a token. These patches are then processed through a series of transformer layers, allowing the model to capture spatial information across the image while maintaining computational efficiency, leveraging the strengths of both vision transformers and convolutional neural networks [35].

The architecture of Swin Transformer is as follows:

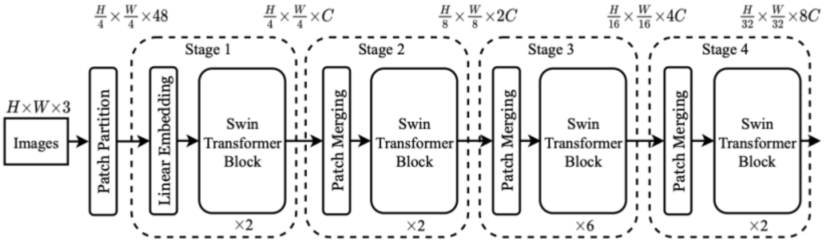


Fig. 3. Swin Transformer architecture

2.2.3 Attention-aware neck

We have placed Coordinate Attention (CA) [36] block in object detection algorithm neck enables the model to focus on relevant spatial positions within the images, thus enhancing the ability to detect smaller targets and reducing false negatives count.

CA block focuses on exploiting the spatial information within feature maps by learning attention weights for each spatial position. Unlike traditional attention mechanisms that operate on channel-wise features, CA attends to the relationships between different spatial positions. It aims to capture long-range dependencies and contextual information within the feature maps, thus enhancing the model's ability to understand spatial relationships and capture fine-grained details.

Firstly, a pair of direction-aware feature maps is generated for two spatial dimensions, using the following transformations:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i)$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w)$$

where z are the outputs of the c-th channel at width w and height h respectively, which are then concatenated and sent to a 1x1 convolutional function F_1 as follows:

$$\mathbf{g}^h = \sigma\left(F_h(\mathbf{f}^h)\right)$$

$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w))$$

where δ is a non-linear activation function and $\boldsymbol{f}$ is the intermediate feature map containing spatial information along horizontal and vertical dimensions. Next, 1x1 convolutional transformations F_h and F_w are applied to f^{h} and f^{w}:

$$\mathbf{g}^h = \sigma\left(F_h(\mathbf{f}^h)\right)$$
$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w))$$

where σ is the sigmoid function. The outputs of this transformations are used as attention weights for both spatial dimensions, yielding final block output:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j)$$

The overall structure of CA block can be shown as follows:

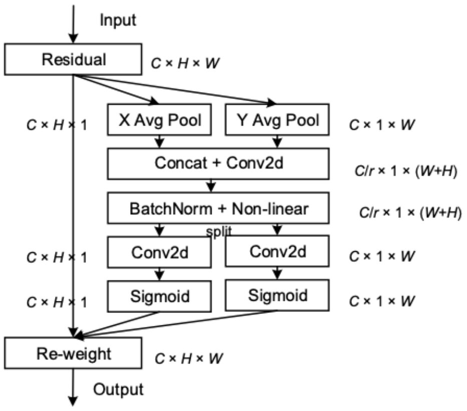


Fig. 4. The structure of Coordinate Attention (CA) block

2.2.4 Separate head for small object detection

In YOLOv8, final detection is one by three detection heads which accept the feature maps of resolution 80 x 80, 40 x 40 and 20 x 20. By conducting experiments with higher resolution feature map, we have come to the conclusion that integrating an extra object detection head for processing 160 x 160 feature maps can greatly improve small object detection performance, by effectively using more global and local contextual information, which gets diluted during the upsampling process, but still present in earlier feature maps.

3. EXPERIMENT RESULTS AND PROPOSED SOFTWARE

3.1 Data overview

A challenging underwater detection dataset UTDAC2020, which is short for Underwater Target Detection Algorithm Competition 2020, has been selected to test the performance of the proposed algorithm. The dataset features 5168 training and 1293 validation images in various resolutions (3840 x 2160, 1920 x 1080, 720 x 405 and 586 x 480), featuring 4 classes (echinus, holothurian, scallop and starfish).

3.2 Implementation details

The experimental setup consisted of Intel Xeon E5 CPU (2.00 GHz), two NVIDIA Tesla T4 GPU with 16GB VRAM each with Ubuntu 20.04.6 LTS installed, Python version 3.10.13, CUDA version 12.1, PyTorch version 2.2.1. The training process was limited with 200 epochs with early stopping implemented, batch size was fixed at 32, stochastic gradient descent has been used as an optimization algorithm with momentum 0.95 and weight decay coefficient 0,005 and initial learning rate 0.01. Default augmentation strategies from YOLOv8 have been applied, and no other augmentations have been used. All the samples from datasets have been processed by image preprocessing module, being resized to 640 x 640 and enhanced by CLAHE algorithm.

3.3 Metrics and experiment results

The following metrics have been used to assess the performance of the algorithm:

- Precision, defined as true positives count, divided by the sum of true positives and false positives, indicating false-detection rate of the algorithm;
- Recall, defined as true positives count, divided by the sum of true positives and false negatives, reflecting the missed-detection rate of the algorithm;

- mAp$^{IoU=0.5}$, defined as the mean average precision (mAp) for all target classes across entire dataset with IoU = 0.5 set as an evaluation threshold;
- Floating point operations count, measured in GFLOPs, reflecting the computational complexity of the network.

Model	**Precision**	**Recall**	**mAp$^{IoU=0.5}$**	**GFLOPs**
YOLOv8n	71.02%	66.92%	82.65%	8.1
YOLOv8s	75.02%	69.78%	84.71%	28.4
YOLOv8m	76.72%	70.53%	84.92%	78.7
YOLOv8l	79.24%	73.12%	85.09%	164.8
[30]	82.71%	80.74%	86.32%	282.05
[31]	-	-	85.49%	25.5
Ours	82.67%	81.02%	86.3%	24.1

Table 2. Results comparison

The proposed model surpassed a much larger YOLOv8l model in precision, recall and mAp$^{IoU=0.5}$ metrics, while maintaining smaller size for deployment.

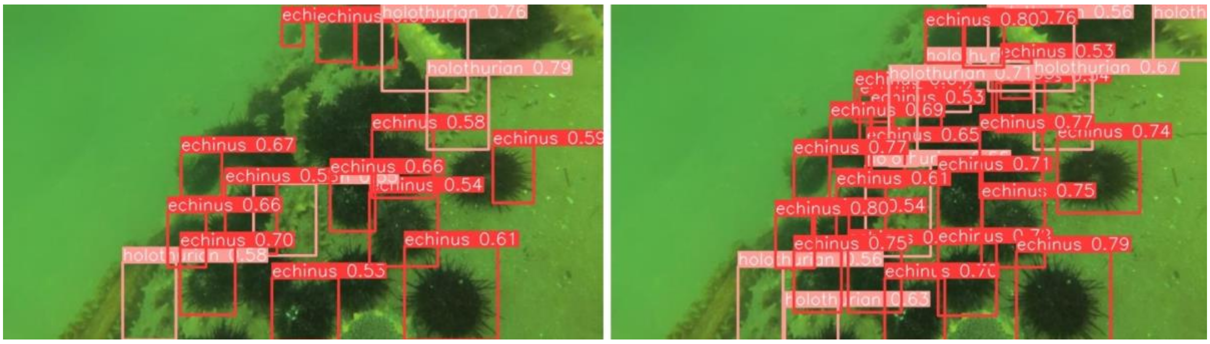


Fig 5. Image from the validation set processed by the baseline YOLOv8 (left) and our model (right). The number of misdetections of small dense objects decreased significantly.

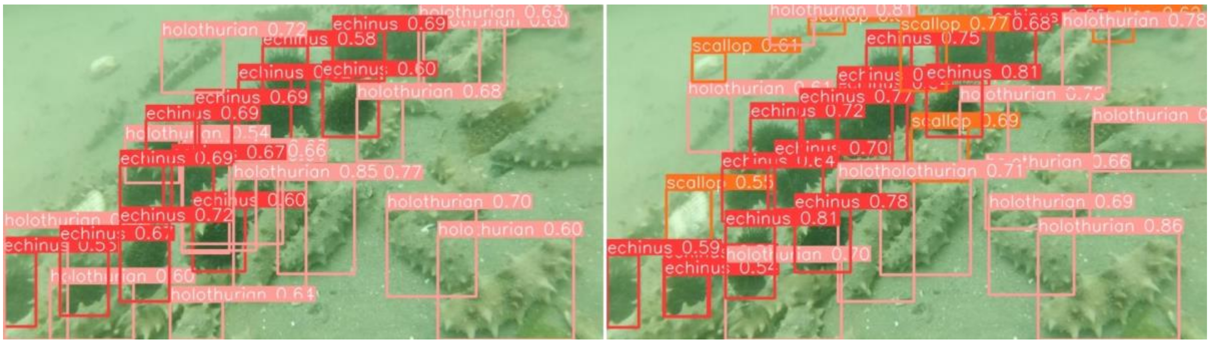
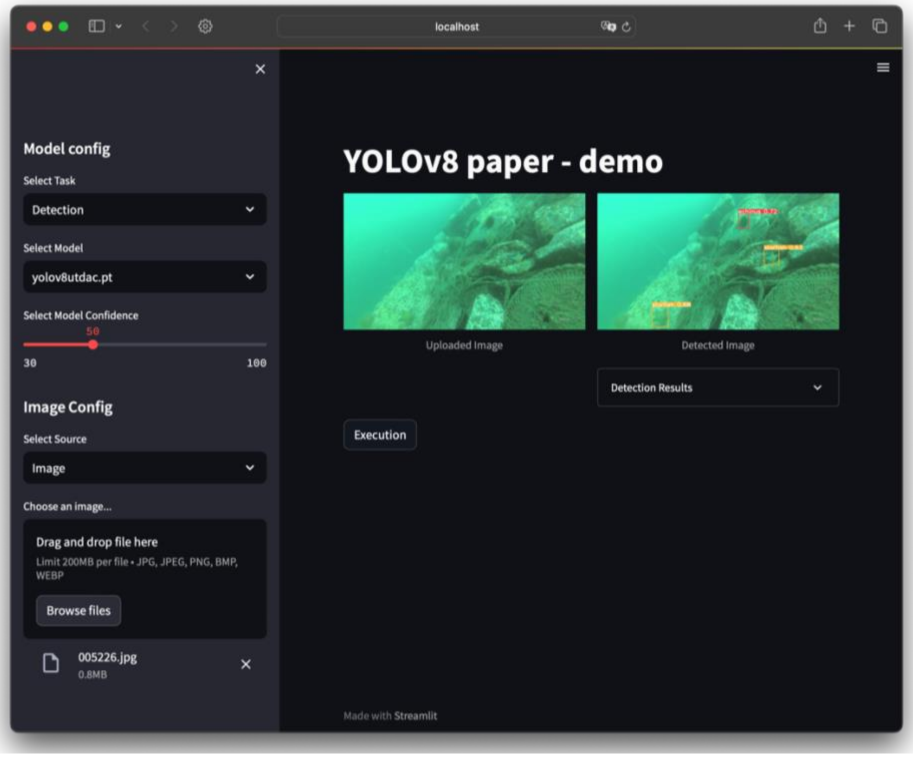


Fig 6. YOLOv8 (left) often missed the targets from underrepresented categories like scallop, while our model (right) detected it successfully

3.4 Proposed software

To better demonstrate our proposed object detection framework results, we have built an interactive front-end. Currently, it supports uploading an image, processing it with modified YOLOv8 network and yielding object detection results.



localhost

Model config

Select Task: Detection

Select Model: yolov8utdac.pt

Select Model Confidence: 50 (30–100)

Image Config

Select Source: Image

Choose an image...

Drag and drop file here

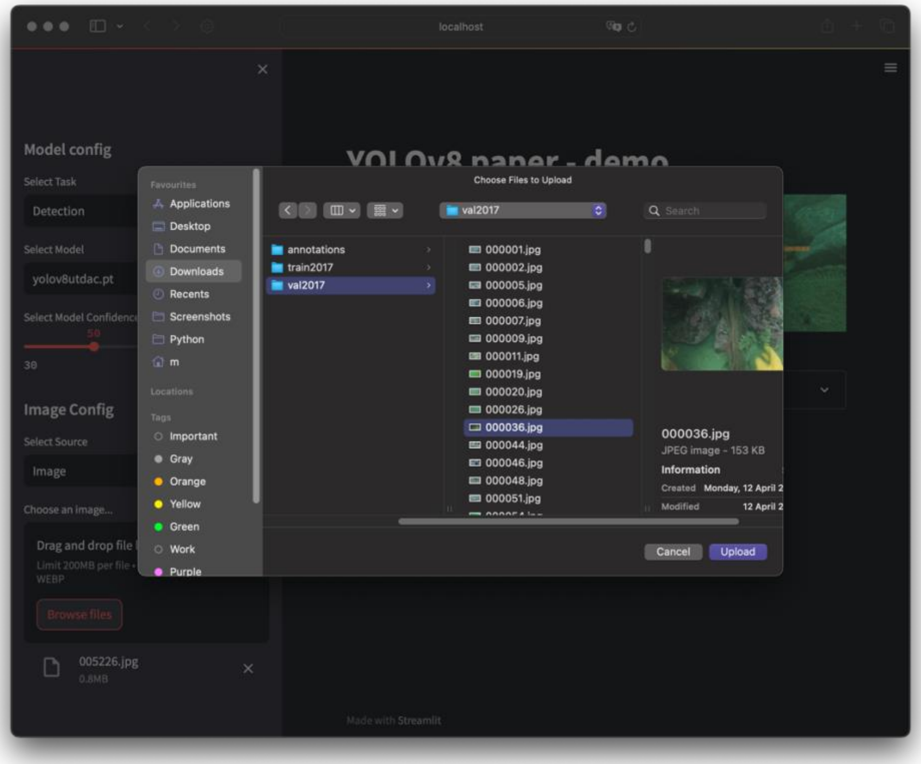
Limit 200MB per file • JPG, JPEG, PNG, BMP, WEBP

005226.jpg 0.8MB

YOLOv8 paper - demo

Made with Streamlit

Fig 7. The interface of proposed software

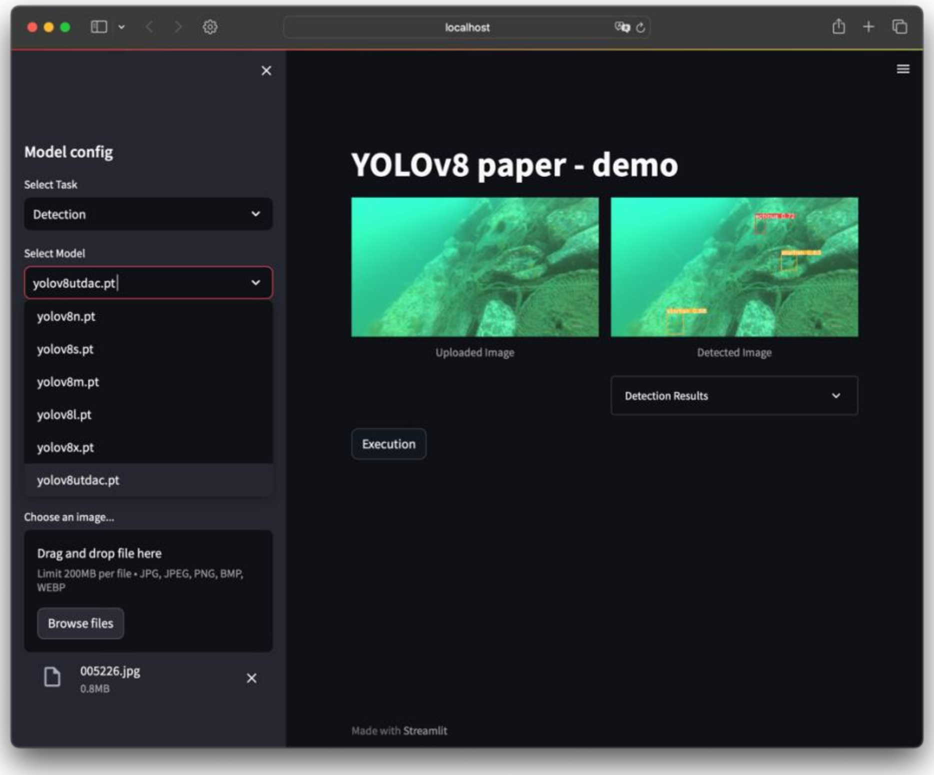


Select Task: Detection

Select Model: yolov8utdac.pt

Select Source: Image

Fig 8. File uploading dialogue window



Select Task: Detection

Select Model: yolov8utdac.pt

Fig 9. Model selection includes pre-packaged versions of YOLO to compare the results

CONCLUSION

This paper proposes an intelligent system tailored for Underwater Object Detection, addressing the intricate challenges inherent to this task. Through implementing a pre-processing module, a series of enhancements to the YOLOv8 architecture, we have significantly improved the detection performance in underwater environments and made the system suitable for deployment.

Firstly, to mitigate visibility issues, cope with small densely packed objects, and handle target occlusion, we introduced a novel image preprocessing module utilizing Contrast Limited Adaptive Histogram Equalization. This preprocessing step enhances the contrast of underwater images, thereby facilitating more accurate object detection. We enhanced the standard CSPDarknet53 backbone of YOLOv8 with transformer block, which improves detection accuracy in challenging underwater conditions, especially with the targets of varying sizes. To effectively highlight regions of interest within the image, we incorporated attention mechanism in object detection neck architecture, which enables the model to focus on crucial features, enhancing the overall detection performance. Additionally, we introduced an extra classification head, which leverages higher resolution feature maps to capture more information about small targets.

Through experimentation on the UTDAC2020 dataset, our model achieves 82.67% precision, 81.02% recall, and 86.3% mean average precision at IoU=0.5. Notably, our framework outperforms the YOLOv8s model by a significant margin, while also being 15.1% smaller in terms of computational complexity. These results underscore the efficacy and efficiency of our proposed framework for underwater object detection tasks, demonstrating its potential for real-world applications in underwater environments.

REFERENCES

[1]F. Alenezi, A. Armghan, and K. C. Santosh, “Underwater image dehazing using global color features,” Engineering Applications of Artificial Intelligence, vol. 116, p. 105489, Nov. 2022, doi: https://doi.org/10.1016/j.engappai.2022.105489.

[2]K. Hu, C. Weng, Y. Zhang, J. Jin, and Q. Xia, “An Overview of Underwater Vision Enhancement: From Traditional Methods to Recent Deep Learning,” Journal of Marine Science and Engineering, vol. 10, no. 2, p. 241, Feb. 2022, doi: https://doi.org/10.3390/jmse10020241.

[3] M. Han, Z. Lyu, T. Qiu, and M. Xu, “A Review on Intelligence Dehazing and Color Restoration for Underwater Images,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 5, pp. 1820–1832, May 2020, doi: https://doi.org/10.1109/tsmc.2017.2788902.

[4]H. Hu, L. Zhao, B. Huang, X. Li, H. Wang, and T. Liu, “Enhancing Visibility of Polarimetric Underwater Image by Transmittance Correction,” vol. 9, no. 3, pp. 1–10, Jun. 2017, doi: https://doi.org/10.1109/jphot.2017.2698000.

[5]Kaiming He, Jian Sun, and Xiaoou Tang, “Single Image Haze Removal Using Dark Channel Prior,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2341–2353, Dec. 2011, doi: https://doi.org/10.1109/tpami.2010.168.

[6]X. Fu, P. Zhuang, Y. Huang, Y. Liao, X.-P. Zhang, and X. Ding, “A retinex-based enhancing approach for single underwater image,” International Conference on Image Processing, Oct. 2014, doi: https://doi.org/10.1109/icip.2014.7025927.

[7]W.-H. Zhang, G. Li, and Z. Ying, “A new underwater image enhancing method via color correction and illumination adjustment,” Visual Communications and Image Processing, Dec. 2017, doi: https://doi.org/10.1109/vcip.2017.8305027.

[8]R. Liu, Z. Jiang, S. Yang, and X. Fan, “Twin Adversarial Contrastive Learning for Underwater Image Enhancement and Beyond,” IEEE transactions on image processing, vol. 31, pp. 4922–4936, Jan. 2022, doi: https://doi.org/10.1109/tip.2022.3190209.

[9]M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, “Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion,” Remote Sensing, vol. 13, no. 22, p. 4706, Nov. 2021, doi: https://doi.org/10.3390/rs13224706.

[10]H. Liu, P. Song, and R. Ding, “WQT and DG-YOLO: towards domain generalization in underwater object detection,” arXiv (Cornell University), Apr. 2020, doi: https://doi.org/10.48550/arxiv.2004.06333.

[11]W. Lin, J.-X. Zhong, S. Liu, T. Li, and G. Li, “ROIMIX: Proposal-Fusion Among Multiple Images for Underwater Object Detection,” arXiv (Cornell University), May 2020, doi: https://doi.org/10.1109/icassp40776.2020.9053829.

[12]X. Sun et al., “Transferring deep knowledge for object recognition in Low-quality underwater videos,” Neurocomputing, vol. 275, pp. 897–908, Jan. 2018, doi: https://doi.org/10.1016/j.neucom.2017.09.044.

[13]S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: https://doi.org/10.1109/tpami.2016.2577031.

[14]W. Liu et al., “SSD: Single Shot MultiBox Detector,” Computer Vision – ECCV 2016, vol. 9905, pp. 21–37, 2016, doi: https://doi.org/10.1007/978-3-319-46448-0_2.

[15]T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2018, doi: https://doi.org/10.1109/tpami.2018.2858826.

[16]Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” Jul. 2018, doi: https://doi.org/10.48550/arXiv.2107.08430.

[17]Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” Sep. 2019, doi: https://doi.org/10.48550/arXiv.1904.01355.

[18]J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” arXiv.org, Jun. 2015, doi: https://doi.org/10.48550/arXiv.1506.02640.

[19]M. Sung, S.-C. Yu, and Y. Girdhar, “Vision based real-time fish detection using convolutional neural network,” OCEANS 2017 - Aberdeen, Jun. 2017, doi: https://doi.org/10.1109/oceanse.2017.8084889.

[20]M. Pedersen, Joakim Bruslund Haurum, R. Gade, and T. B. Moeslund, “Detection of Marine Animals in a New Underwater Dataset with Varying Visibility,” Computer Vision and Pattern Recognition, pp. 18–26, Jun. 2019.

[21]M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, “Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion,” Remote Sensing, vol. 13, no. 22, p. 4706, Nov. 2021, doi: https://doi.org/10.3390/rs13224706.

[22]L. Chen, Y. Yang, Z. Wang, J. Zhang, S. Zhou, and L. Wu, “Lightweight Underwater Target Detection Algorithm Based on Dynamic Sampling Transformer and Knowledge-Distillation Optimization,” Journal of Marine Science and Engineering, vol. 11, no. 2, pp. 426–426, Feb. 2023, doi: https://doi.org/10.3390/jmse11020426.

[23]K. Liu, L. Peng, and S. Tang, “Underwater Object Detection Using TC-YOLO with Attention Mechanisms,” Sensors, vol. 23, no. 5, p. 2567, Jan. 2023, doi: https://doi.org/10.3390/s23052567.

[24]X. Shen, X. Sun, H. Wang, and X. Fu, “Multi-dimensional, multi-functional and multi-level attention in YOLO for underwater object detection,” Neural Computing and Applications, vol. 35, no. 27, pp. 19935–19960, Jul. 2023, doi: https://doi.org/10.1007/s00521-023-08781-w.

[25]F. Xu, H. Wang, J. Peng, and X. Fu, “Scale-aware feature pyramid architecture for marine object detection,” Neural Computing and Applications, vol. 33, no. 8, pp. 3637–3653, Jul. 2020, doi: https://doi.org/10.1007/s00521-020-05217-7.

[26]T.-S. Pan, H.-C. Huang, J.-C. Lee, and C.-H. Chen, “Multi-scale ResNet for real-time underwater object detection,” Signal, Image and Video Processing, vol. 15, no. 5, pp. 941–949, Nov. 2020, doi: https://doi.org/10.1007/s11760-020-01818-w.

[27]A. A. Muksit, F. Hasan, Md. F. Hasan Bhuiyan Emon, M. R. Haque, A. R. Anwary, and S. Shatabda, “YOLO-Fish: A robust fish detection model to detect fish in realistic

underwater environment," Ecological Informatics, vol. 72, p. 101847, Dec. 2022, doi: https://doi.org/10.1016/j.ecoinf.2022.101847.

[28]Long Qing Chen et al., "Underwater object detection using Invert Multi-Class Adaboost with deep learning," Jul. 2020, doi: https://doi.org/10.1109/ijcnn48605.2020.9207506.

[29]Long Qing Chen et al., "Underwater object detection using Invert Multi-Class Adaboost with deep learning," Jul. 2020, doi: https://doi.org/10.1109/ijcnn48605.2020.9207506.

[30]Z. Wang, G. Zhang, K. Luan, C. Yi, and M. Li, "Image-Fused-Guided Underwater Object Detection Model Based on Improved YOLOv7," Electronics, vol. 12, no. 19, pp. 4064–4064, Sep. 2023, doi: https://doi.org/10.3390/electronics12194064.

[31]M. Zhang, Z. Wang, W. Song, D. Zhao, and H. Zhao, "Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network," Applied Sciences, vol. 14, no. 3, p. 1095, Jan. 2024, doi: https://doi.org/10.3390/app14031095.

[32]Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, pp. 12993–13000, Apr. 2020, doi: https://doi.org/10.1609/aaai.v34i07.6999.

[33]X. Li et al., "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," Jun. 2020, doi: https://doi.org/10.48550/arXiv.2006.04388.

[34]S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas, and K. E. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," in IEEE Xplore, May 1990, pp. 337–345. doi: https://doi.org/10.1109/VBC.1990.109340.

[35]Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," arXiv:2103.14030 [cs], Mar. 2021, Available: https://arxiv.org/abs/2103.14030

[36]Q. Hou, D. Zhou, and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," Mar. 2021, doi: https://doi.org/10.48550/arXiv.2103.02907.