# Deep Learning for Underwater Object Detection: From CNNs to Transformer-Based Real-Time Solutions

Hari Bhandari[1] and Pengcheng Liu[1]

***Abstract*—This paper presents a comprehensive review of recent advancements in underwater object detection (UOD), with a focus on the transformative role of deep learning approaches, particularly Convolutional Neural Networks (CNNs) and Transformer-based architectures. It examines the progression from traditional thresholding and sonar-based techniques to modern, data-driven models capable of addressing the unique challenges posed by underwater environments, including poor visibility, data scarcity, and computational constraints. Special attention is given to the development of hybrid CNN-Transformer models and the integration of sonar and optical data for enhanced detection accuracy. Additionally, the paper highlights the importance of specialised datasets, real-time performance considerations, and ethical implications in deploying AI systems for marine applications. Ongoing research directions are discussed, emphasising the need for efficient, robust, and adaptable models suitable for real-world underwater tasks.**

***Index Terms*—Deep learning, underwater, object detection, CNNs, transformer.**

## I. Introduction

Underwater object detection (UOD) has evolved significantly over the past few decades, transitioning from simple manual inspections and rule-based algorithms to sophisticated deep learning and Transformer-driven approaches. This evolution reflects the growing need for robust, efficient, and accurate detection systems capable of operating in complex marine environments. Applications of UOD are wide-ranging, encompassing ecological monitoring, underwater exploration, marine resource management, and aquaculture.

Early object detection techniques, such as thresholding and traditional model-based strategies, relied heavily on predefined heuristics and manual feature engineering. These methods performed adequately in controlled or predictable environments but struggled under the dynamic and visually challenging conditions of underwater settings. Factors such as poor visibility, light attenuation, turbidity, and occlusion frequently undermine the reliability of these classical approaches.

The advent of deep learning, particularly Convolutional Neural Networks (CNNs), marked a transformative milestone in object detection. CNN-based methods demonstrated an ability to automatically learn hierarchical features from raw image data, greatly surpassing the limitations of handcrafted features. These advancements enabled more robust detection capabilities even in complex underwater scenes. Further innovations, including region-based CNN frameworks and real-time single-shot detectors like YOLO and SSD, offered improved accuracy and speed, opening new possibilities for practical deployment in marine applications.

More recently, Transformer-based architectures have revolutionised the field of computer vision by introducing global self-attention mechanisms that capture long-range dependencies within images. Vision Transformers (ViT), Detection Transformers (DETR), and hybrid CNN-Transformer models have begun to address some of the persistent challenges in UOD, such as detecting objects in low-contrast, cluttered, or noisy underwater environments. Additionally, these architectures offer promising solutions for fusing multimodal data, such as sonar and optical imagery, enhancing detection performance and reliability in real-world scenarios.

Despite these advancements, several challenges remain. Underwater environments impose significant constraints on data acquisition, computational resources, and model generalisation. Limited availability of large-scale, high-quality annotated datasets further complicates the training and evaluation of robust models. Additionally, real-time detection requirements in underwater robotics and autonomous systems necessitate computationally efficient solutions that balance accuracy with speed. This paper presents a comprehensive examination of the progression of underwater object detection technologies. It traces the journey from early heuristic-based methods to state-of-the-art CNNs and Transformer-based models. Furthermore, it highlights key challenges, ongoing research efforts, and emerging trends aimed at developing efficient and adaptable detection systems suitable for deployment in complex and resource-constrained underwater environments.

## II. Early Approaches & Sonar

### A. Manual Methods & Thresholding

Early object detection, particularly in industrial and surveillance applications, relied on manual inspections or simplistic automated routines [1]. In assembly lines, for example, human inspectors or technicians would visually scan products for apparent defects [1]. This approach was economical when the environment was steady; it failed as scenes grew more complex or inspection periods prolonged. Initial algorithmic

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/Y000773/1. (Corresponding author: Pengcheng Liu)

[1]Hari Bhandari and Pengcheng Liu are with the Department of Computer Science, University of York, YO10 5GH, United Kingdom. E-mails: {hb1622, pengcheng.liu}@york.ac.uk

methods typically used threshold-based rules to identify significant deviations from a baseline image or background. These techniques were efficient when the environment was uniformly lit and had low clutter but they quickly lost effectiveness in dynamic conditions or when objects partially overlapped [2]. This is very normal and expected in underwater settings.

In underwater settings, threshold-based approaches are particularly challenging [3]. They operate as crude filters that capture only the most obvious outliers.Light absorption, colour changes, and turbulence cause abrupt and random changes in pixel brightness, which makes it impossible to use a single global threshold. Even slight changes in the depth or water transparency can greatly alter the appearance of an object, which can lead to many false alarms when the background noise exceeds the threshold or missed detections when the background is dark and objects are indistinguishable from it because of low contrast [3]. This highlights the need for more intelligent, learning-based strategies for object detection in underwater environments.

### *B. Sonar for Underwater Exploration*

Historically, UOD has relied on sonar for so many years [4]. Sonar systems work by transmitting acoustic pulses and measuring the returning echoes to map the seafloor, find large aquatic animals, or follow subsea structures [4]. Although sonar is good for coarse localisation in even the darkest or murkiest waters, it cannot typically resolve subtle differences between organisms or detect small scale anomalies. This limitation has in turn led to the consideration of camera-based detection in underwater research, for example, for identifying specific fish species in aquaculture facilities. However, reconciling sonar readings with optical imagery involves complex data fusion processes, due to significant differences between the two modalities [4].

### *C. Feature-Based Computer Vision*

Before the advent of deep learning, researchers turned to more advanced feature-based algorithms [5]. Approaches like edge detection, optical flow, and background subtraction improved upon raw thresholding by isolating motion or well-defined object boundaries. Eventually, descriptors such as Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) allowed more robust detection by capturing gradients and corners across different scales or angles [6], [7]. These pipelines coupled with machine learning classifiers including Support Vector Machines (SVM) did better in different conditions but they were still a far cry from being independent and required extensive feature engineering [5]. These algorithms were often rendered less helpful in underwater contexts by subtle shifts in colour or illumination, resulting in frequent misclassifications unless they were meticulously tuned.

## III. Deep Learning in Object Detection

### *A. LeNet-5: Foundations*

The field of object detection has seen a remarkable growth primarily driven through the advancements in CNNs [8]. CNNs gained recognition through the work of Yann LeCun and collaborators, particularly on LeNet-5 [9]. It was initially designed for handwritten number recognition 0,1..9 but demonstrated significant potential of CNNs . LeNet-5 demonstrated that networks featuring convolutional and pooling layers could learn relevant features from pixels directly. This marked a clear improvement over traditional handcrafted feature extraction methods. However, early CNNs had notable challenges, including vanishing gradients—where deeper layers failed to update properly during training—high computational requirements, and a tendency to overfit, especially when datasets were small [8].

### *B. AlexNet & Advances*

The transition from LeNet-5 to AlexNet was motivated by the limitations of shallow networks [10]. The complexity of datasets, increasing demands for accuracy, and the advancement of GPU hardware led researchers to pursue deeper, more capable CNN architectures. AlexNet [10], notably won the ImageNet competition by leveraging GPU acceleration, using ReLU activations to mitigate the vanishing gradient problem, and incorporating dropout to reduce overfitting [11]. This highlighted the advantages of deeper architectures and established a new baseline for visual recognition tasks. Subsequent architectures, including ZFNet [12], GoogLeNet [13], VGGNet [14], and ResNet [15], expanded further, refining network depth, kernel dimensions, and connection patterns to improve performance across various computer vision tasks beyond simple classification; they could also be adapted for tasks like segmentation and object detection, fueled by bounding-box regression and region proposal strategies [16].

## IV. Region-Based CNNs

### *A. From R-CNN to Faster R-CNN*

Region-Based CNN (R-CNN) represented a turning point in object detection. Initially, R-CNN [17] generated bounding-box proposals through selective search, then employed a CNN to classify each proposal. Although significantly more accurate than earlier techniques, the process was computationally intensive since each proposal required a separate pass through the network. Fast R-CNN addressed this inefficiency by sharing convolutional computations and using a region of interest pooling strategy, thereby accelerating inference. Faster R-CNN further refined this approach through the introduction of a Region Proposal Network (RPN), which produced bounding-box suggestions with minimal overhead [18]. Faster R-CNN went on to establish leading performance on well-known datasets like Pascal VOC and MS COCO, blending high accuracy with reasonable speed.Region-based CNNs did not remain confined to terrestrial imagery. A study by Han et al. [19] illustrated the effectiveness of Faster R-CNN in classifying marine wildlife captured in underwater video. Despite uneven lighting and challenging water conditions, they achieved a mean Average Precision (mAP) of 74.63 at an IoU threshold of 0.5. This demonstrated how region-based methods, when trained with domain-specific data, could handle many of

the complexities typically encountered in marine ecosystems. Sonar might remain valuable for broader-scale surveys, but region-based CNNs excelled at identifying individual fish or other organisms with fine-grained accuracy [19] .

## V. Single-Shot & Real-Time Frameworks

### A. Speed in Detection

Although region-based models reached high precision, their multi-stage workflows often limited real-time viability. In contexts such as autonomous driving or continuous underwater footage analysis, inference speed becomes critical as we are computationally limited. Single-shot detectors, combining bounding-box prediction and classification into one super efficient step [20].

### B. SSD: Single Shot MultiBox Detector

The Single Shot MultiBox Detector (SSD) introduced by [20] marked a massive advancement in object detection especially in real-time. It captures images at multiple scales and uses several feature maps to do so, each of which is responsible for the detection of objects of different sizes. SSD achieved this using convolutional predictors on these multi-scale feature maps, therefore eliminating the need for separate proposal generation stages, improving both speed and accuracy [20]. These qualities made it very suitable for application in real-time underwater detection tasks, where quick and accurate decisions are very important.

### C. YOLO Iterations

YOLO (You Only Look Once) [21] sets the stage by treating detection more like a regression problem. It divides the input image into grids. Each cell in the grid predicts bounding boxes and class probabilities simultaneously all within single forward pass. This design improved speed considerably, although initial versions struggled with smaller objects or heavy clutter. This modal was designed to be simple yet very efficient in interference and extremely suitable for real time application. Over time, refinements like YOLOv3 [22], YOLOv5 [23] and all the way up recently released YOLOv12 [24] incorporated advanced data augmentation, more pronounced robust backbones, and refined anchor box mechanisms to narrow the gap between single-stage and region-based detectors in terms of accuracy [24]. These developments have cemented the position of YOLO as crucial in real-time underwater monitoring tasks where objects need to be detected quickly and accurately for duties such as aquaculture monitoring and ecological surveys.

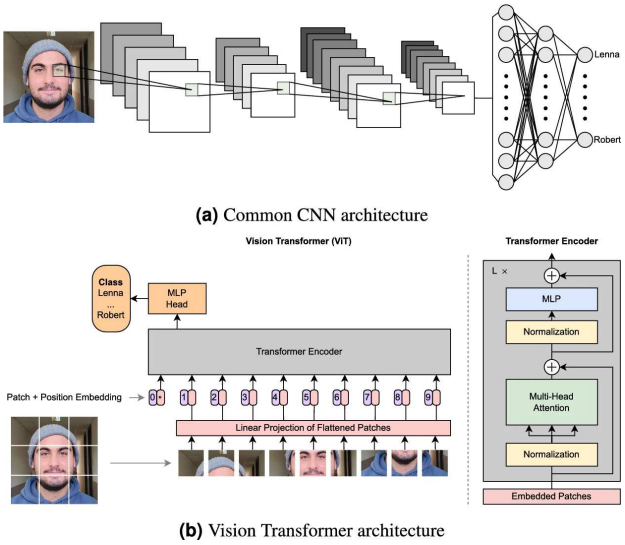
## VI. Underwater Detection: Sonar & Optical

The Sonar remains essential for large scale underwater mapping but lacks detailed identification capabilities, unlike optical imaging, which excels at capturing fine morphological features. Kim et al. [25] improved underwater detection accuracy by fusing sonar with optical imagery, significantly enhancing image clarity. However, reconciling geometric differences between these modalities remained challenging due to geometric calibration, noise reduction, and real-time fusion coordination of these distinct data types.

## VII. Transformers in Vision

### A. Self-Attention

Transformers first made massive impact in natural language processing with openai's GPT series. It completely changed the way we think about AI by allowing models to understand context from anywhere within a sentence. When GPT came out, it dramatically shifted expectations, demonstrating the potential of global self-attention [26]. Researchers at google asked this question,If transformers works for a sequence of words, can we treat images like sequences too? [27]Not long after, these ideas jumped into the computer vision world through Vision Transformers (ViT), which apply the same principle by splitting images into patches and using attention mechanisms to understand global relationships quickly [27]. CNNs rely on local convolutions and pooling layers to extract hierarchical features. In contrast, Transformers split the image into patches and utilise global self-attention (as shown in Figure 1) Now, these powerful ideas have found their way underwater, inspiring new models like DETR and Swin Transformer, which are tackling the unique challenges of detecting objects in complex marine environments [28].




**(a)** Common CNN architecture

**(b)** Vision Transformer architecture

Fig. 1. Comparison of common CNN architecture (top) and Vision Transformer (ViT) architecture (bottom) Adapted from [29]. CNNs extract features through stacked convolution and pooling layers, while Transformers operate on embedded patches using global self-attention.

### B. DETR, Swin, and RT-DETR

Building on the success of ViT, Detection Transformer (DETR) further advanced object detection by reformulating it as a set prediction problem, eliminating traditional bounding-box proposals and post-processing techniques like non-maximum suppression (NMS) [28]. Although DETR provided high accuracy, it required extensive training time and large annotated datasets. To address this limitation, the Swin Transformer introduced a hierarchical, window-based attention mechanism that drastically reduced computational requirements, allowing models to efficiently scale to higher-resolution

images. More recently, Real-Time DETR (RT-DETR) was developed to bridge the gap between high accuracy and real-time inference speeds which was previously lacking, crucial for practical applications in environments such as underwater monitoring. RT-DETR maintains the benefits of Transformer architectures, including global context understanding, while also being optimised for speed and computational efficiency [30].RT-DETR introduces a hybrid encoder that splits attentions within scales and across scales to reduce computation needed, and super efficient query selection for decoding. This efforts made RT DETR truly the first transformer to rival one stage CNN's in runtime [30]. This makes RT-DETR particularly suited for deployment in resource constrained underwater devices where real-time detection is essential.

| Model | Type | MAP50 | FPS |
|---|---|---|---|
| Faster R-CNN | Two-Stage | 38.5 | 23.9 |
| SSD300 | Single-Stage | 74.3 | 59 |
| YOLOv8n | Single-Stage | 37.3 | 160 |
| ViTDet (Base) | Transformer | 51.1 | N/A |
| RT-DETR-R50 | Transformer | 53.1 | 108 |

TABLE I
PERFORMANCE COMPARISON OF SELECTED OBJECT DETECTION MODELS ON BENCHMARK DATASETS.

## VIII. HYBRID METHODS

### A. CNN & Transformer Integration

Hybrid models attempt to merge CNN-based local feature extraction with the global self-attention of Transformers. For instance, a CNN might quickly encode lower-level shapes or textures, followed by a Transformer stage that refines bounding boxes using distant correlations in the image. Preliminary studies indicate that such hybrids can surpass pure CNN or Transformer setups in dense object scenarios. Nevertheless, optimising such systems is challenging since two totally different learning paradigms must be reconciled [35], [36].

### B. Ensemble Methods

Ensemble methods have been picking up in popularity over last few years. It works by combining the outputs from multiple architectures. For eg. predictions from transformer based can be integrated with predictions from CNN based detector. As shown on this study, these methods increase the performance but also increase the computational complexity. This very often limits ensembles to more offline analysis tasks like processing footage of coral reefs to document biodiversity. It would be not suited for real time scenarios especially where resources are limited like underwater [37].

## IX. VIDEO TRACKING TECHNIQUES

### A. Classical Tracking

When object detection moves into multi-frame contexts, an additional layer of complexity arises. Traditional video trackers, including feature-based (SIFT, HOG key points) and model-based (shape fitting over time), attempt to maintain object identities from one frame to the next. Underwater environments, however, amplify the risk of occlusion by sediment or algae and can involve irregular lighting shifts caused by currents. Classic trackers frequently require elaborate parameter tuning to remain stable in such fluid conditions. [38]

### B. Deep Learning Tracking

Recent developments have seen deep learning methods widely adopted for video tracking, significantly overcoming limitations of classical techniques. Deep learning-based tracking combines more sophisticated CNN or Transformer-based detectors with re-identification (ReID) methods [39]. With that, It is providing consistent object tracking across multiple video frames. These methods automatically learn robust features. Therefore reducing reliance on manual parameter adjustments. While more computationally intensive, these systems greatly enhance the monitoring capabilities of marine biologists and aquaculture managers by providing continuous, reliable tracking. This technology has proven particularly beneficial for detailed studies of marine animal behaviour, monitoring fish health, and even early detection of issues within aquaculture facilities which is greatly helpful.

## X. DATASETS & UNDERWATER DATA

### A. Mainstream Datasets

There are few widely used datasets like MS COCO, Pascal VOC, and Open Images that have dominated object detection research, providing diverse categories and ground-truth annotations. They act as universal benchmarks that allow model comparisons. Yet, these collections rarely account for the environmental shifts or specialised objects found underwater. Models trained on them often struggle with the as underwater differs significantly from standard images: colours are shifted (blue/green dominance as red light attenuates quickly), visibility is mostly poor and backgrounds can be extremely complex (eg. coral reefs) or even open water [2].

### B. Domain-Specific Collections

Since underwater environments come with unique challenges, specialised datasets have been created to account for changing water clarity, shifting light conditions, and interactions between different species. Notable examples include, Fish4Knowledge [40], URPC [41], Brackish [42]. These datasets usually contain labelled images of deepwater fish, algae, and even man-made structures like cages or mooring lines. While they are essential for training models that work well in aquatic settings, they often have limitations—such as being small in size or having class imbalances, where common fish species appear frequently while rarer ones are underrepresented.

[0]Data sourced from COCO benchmarks and official repositories based on the reported benchmarks [31]–[34].

### C. Training Implications

When AI models are trained on images from land environments, they often struggle with underwater scenes, meaning they need extra adjustments to work properly. One way to help is through data augmentation, which involves adding effects like colour shifts or noise to make the images resemble real underwater conditions. However, fully adapting these models to marine environments is rarely straightforward. Since collecting and labelling underwater data takes a lot of time and money, collaboration between researchers and marine scientists is essential to combine their expertise and improve model accuracy [2].

## XI. Bias, Fairness & Ethics

### A. Sources of Bias

Bias in the training dataset is a key issue that can cause the AI models to emit biased predictions [43]. For example, in 2015-2017, Volvo tested its self-driving car in Australia, which was trained on European wildlife. The system was found to have a problem with distinguishing between kangaroos, due to their hop, not run, which affected the tracking and detection [44]. This evidenced the critical existing problem of insufficient diversity in the datasets. Similarly in aquaculture, a model trained on data from one region might do well at spotting local issues but fail to detect diseases or conditions that are more common elsewhere [45].

### B. Real-World Impact

Such bias has a profound impact in real-world It can skew estimates of fish populations, hinder invasive species detection, misclassify objects or affect feeding strategies that are applied in different parts of the world [43]. These problems have ecological as well as commercial consequences if decision-makers rely heavily on automated detections. Ethical conversations in the field of artificial intelligence have, for the most part, concentrated on issues directly impacting humans. However, this focus is beginning to shift. They are now gradually expanding to include environmental implications. Ensuring balanced coverage of species and habitats thus remains a priority. Achieving fair representation of species or classes dealing with remains a critical concern [43].

### C. Mitigation Strategies

Studies have been conducted in practices such as selective data collection, synthetic over-sampling of the minority classes, and repeated validation of the models using iterative approaches. It is therefore more important than ever to provide evidence of the characteristics of the dataset and to regularly assess the performance of the model [2]. This is even more important in high-stakes domains where there is little or no room for error such as marine conservation and aquaculture.

## XII. Performance Metrics & Constraints

### A. Accuracy Metrics

Mean Average Precision (mAP) is widely recognised as the benchmark for object detection accuracy at various Intersections over Union (IoU) thresholds. Evaluations sometimes distinguish performance on small, medium, or large objects, revealing whether certain scales pose recurring issues. In multi-frame scenarios, metrics like Multiple Object Tracking Accuracy (MOTA) may be added to assess the consistency of detection across consecutive frames. [46]

### B. Speed & Resources

In real-time applications, performance is typically assessed using metrics such as frames per second (FPS) or inference time in milliseconds (ms). If a system processes fewer than 10–15 FPS, it might not work well for fast-moving situations. Nevertheless, this isn't a big problem for offline or pre-recorded video analysis. Hardware choice also makes a big difference: Transformers typically demand greater memory and computational resources compared to CNNs, which can often be simplified through pruning or quantization to run smoothly on smaller devices [47]. It becomes more tricky underwater as it brings additional constraints, such as battery life limited data transmission rates, and limited computational power that require lighter, more specialised models that deliver efficiency without compromising performance [37].

### C. Underwater Adaptation

Underwater detection models have to be adaptive to variability in water clarity, lighting, and environment including currents and depth. A model that works well in one situation might not perform the same in another unless it is retrained or adjusted [48]. This is why it is important to test the model extensively in various marine conditions. Fish farms provide stable conditions for controlled environments, but open ocean deployments require far greater adaptability as conditions cans shift quite dramatically between day time or night time or at varying depths.

## XIII. Conclusions

This paper has outlined the progression of UOD technologies, from early manual and threshold-based methods to advanced deep learning and Transformer-based models. CNNs significantly improved detection accuracy, while real-time frameworks like YOLO and SSD enabled practical deployment in dynamic underwater environments. More recently, Transformer-based models, such as DETR and Swin Transformers, have further enhanced detection by incorporating global context understanding. More recent hybrid architectures combining CNNs and Transformers offer promising solutions by leveraging the strengths of both paradigms, balancing local feature extraction with global context understanding.

Despite these advancements, challenges remain, including limited annotated datasets, computational constraints in underwater systems, and the need for robust models adaptable to varying marine conditions. Future research should focus on

multimodal sensor fusion, domain adaptation, and lightweight architectures to enable efficient and accurate underwater object detection for real-world applications.

## REFERENCES


[1] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[2] M. Jian, N. Yang, C. Tao, H. Zhi, and H. Luo, "Underwater object detection and datasets: a survey," *Intelligent Marine Technology and Systems*, vol. 2, no. 1, p. 9, 2024.

[3] P. Singh, B. B. V. L. Deepak, T. Sethi, and M. D. P. Murthy, "Real-time object detection and tracking using color feature and motion," in *Proceedings of the 2015 International Conference on Communications and Signal Processing (ICCSP)*, Melmaruvathur, India, 2015, pp. 1236–1241.

[4] H. K. Alaie and H. Farsi, "Passive sonar target detection using statistical classifier and adaptive threshold," *Applied Sciences*, vol. 8, no. 1, p. 61, 2018.

[5] M. Gordan, O. Dancea, I. Stoian, A. Georgakis, and O. Tsatos, "A new svm-based architecture for object recognition in color underwater images with classification refinement by shape descriptors," in *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, Cluj-Napoca, Romania, 2006, pp. 327–332.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] A. Chatterjee, "Evolution of cnn architectures: Lenet, alexnet, zfnet, googlenet, vgg and resnet," https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848CNN Architectures — LeNet, AlexNet, VGG, GoogLeNet and ResNet — by Prabhu Raghav, 2018.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012, pp. 1097–1105.

[11] M. Team, "Alexnet: The first cnn to win imagenet," https://www.mygreatlearning.com/blog/alexnet-the-first-cnn-to-win-image-net/, January 2025.

[12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[17] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 28, 2015, pp. 91–99.

[19] F. Han, J. Yao, H. Zhu, and C. Wang, "Underwater image processing and object detection based on deep cnn method," *Journal of Sensors*, vol. 2020, p. 6707328, 2020.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525.

[23] G. Jocher, "YOLOv5: Implementation of YOLO object detector," https://github.com/ultralytics/yolov5, 2020.

[24] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," *arXiv preprint arXiv:2502.12524*, Feb. 2025.

[25] H.-G. Kim, J. Seo, and S. M. Kim, "Underwater optical-sonar image fusion systems," *Sensors*, vol. 22, no. 21, p. 8445, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8445

[26] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI*, 2018.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 213–229.

[29] M. Rodrigo, C. Cuevas, and N. García, "Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks," *Scientific Reports*, vol. 14, 2024.

[30] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "DETRs beat YOLOs on real-time object detection," *arXiv preprint arXiv:2304.08069*, 2024.

[31] Ultralytics, "Yolov8 official documentation - performance metrics," 2024.

[32] I. O. Toolkit, "Faster r-cnn resnet101 coco sparse 60 - model zoo readme," 2024.

[33] V.-T. Authors, "Vitdet - vision transformer for dense prediction," 2024, accessed: 2025-03-23. [Online]. Available: https://github.com/ViTAE-Transformer/ViTDet

[34] Ultralytics, "Rt-detr official documentation - overview and performance," 2024.

[35] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, "Lightweight underwater object detection based on YOLOv4 and multi-scale attentional feature fusion," *Remote Sensing*, vol. 13, no. 22, p. 4706, 2021.

[36] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 10 012–10 022.

[37] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings of the International Conference on Image Processing (ICIP)*, 2016, pp. 3464–3468.

[39] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[40] R. B. Fisher, "Fish4knowledge dataset," https://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/GROUNDTRUTH/, 2012, accessed: 2025-03-23.

[41] U. R. Competition, "Urpc2019 dataset," https://universe.roboflow.com/underwater-fish-f6cri/urpc2019-nrbk1, 2019, accessed: 2025-03-23.

[42] M. Pedersen, H. Midtiby, and P. Christiansen, "Brackish dataset," https://www.kaggle.com/aalborguniversity/brackish-dataset, 2019, accessed: 2025-03-23.

[43] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1521–1528.

[44] World Economic Forum, "Kangaroos are confusing self-driving cars," *World Economic Forum*, 2017.

[45] M. S. Ahmed, T. T. Aurpa, and M. A. K. Azad, "Fish disease detection using image based machine learning technique in aquaculture," *arXiv preprint arXiv:2105.03934*, 2021.

[46] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 2012, pp. 1097–1105.

[48] C. Li, J. Guo, R. Cong, Y. Pang, and B. Wang, "Underwater image enhancement by dehazing with minimum information loss and histogram distribution prior," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5664–5677, 2016.