Underwater Object detection

Akshata Kumble
Dept. Electronics and Communication
PES University
Bangalore, India
akshatapraveenkumble@pesu.pes.edu

Sriveda Medatati
Dept. Electronics and Communication
PES University
Bangalore, India
srivedamedatati@pesu.pes.edu

Shruthi M L J
Dept. Electronics and Communication
PES University
Bangalore, India
shruthimlj.pes.edu

***Abstract*—Underwater object detection is a challenging task in computer vision, as traditional methods may struggle to detect objects reliably due to the high levels of background noise and clutter present in underwater environments. To address this problem, we propose an underwater object detection method based on the different algorithms such as "You Only Look Once" (YOLO), Convolutional Neural Networks (CNN), GoogleNet and ResNet. Our approach aims to balance accuracy and speediness for target detection in marine environments, specifically targeting fishes, pipelines, rocks, and other obstacles that could impede the functioning of submarines. We trained and evaluated these models on a dataset of underwater images and observed that YOLO gives us the best results for real time underwater object detection. The results show that our proposed approach is effective for underwater object detection and could have applications in marine biology, oceanography, and underwater exploration.**

***Index Terms*—YOLO, underwater object detection, deep learning, CNN**

I. Introduction

Underwater object detection is a challenging and attractive task in computer vision with various applications in marine biology, oceanography, and underwater exploration. In recent years, significant progress has been made in object detection using artificial intelligence, machine learning, and deep learning techniques. However, detecting objects in underwater environments remains a challenge due to high levels of background noise and clutter, as well as the need for reliable and efficient detection algorithms. Autonomous underwater vehicles, such as submarines, face obstacles in their path that could hamper their functioning, highlighting the need for accurate and reliable underwater object detection methods.

To address this challenge, we propose an underwater object detection method based on the You Only Look Once (YOLO) v5 algorithm. YOLOv5 is a state-of-the-art deep learning model that has shown promising results for object detection tasksin various domains, including computer vision. Our approach seeks to accomplish an optimal balance of accuracy and speed for detecting targets in aquatic environments. We consider targets such as fishes, pipelines, rocks, and other hurdles that could hamper the functioning of submarines underwater.

The YOLO method is important for the following reasons: it can forecast objects in real-time, that enhances detection speed and provides accurate results with minimal background errors. The YOLO algorithm also has great learning capabilities, enabling it to learn object representations and apply them to object detection.

II. Literature Survey

In this section, we review some of the recent works related to underwater object detection, with a focus on those that have used YOLOv5 and other deep learning models. [2] proposed a deep learning approach for underwater object detection using a variant of the YOLOv3 model. The authors used transfer learning to fine-tune the pre-trained model on an underwater dataset, achieving a mean average precision (mAP) of 0.92. They also compared their approach with other state-of-the-art object detection models and found that YOLOv3 outperformed the others in terms of accuracy and speed. [3] proposed a lightweight object detection model based on YOLOv3-tiny for underwater environments. They used a dataset of underwater images containing various objects, including fish, corals, and man-made structures, to train and evaluate their model. The results showed that their approach achieved a high accuracy while maintaining a low computational cost, making it suitable for real-time object detection in underwater environments. [4] proposed an improved version of YOLOv3 for underwater object detection by incorporating a spatial pyramid pooling (SPP) module and a channel attention mechanism into the network. They evaluated their approach on an underwater dataset and achieved mAP of 0.88, outperforming the original YOLOv3 model. Recently, YOLOv5 has gained popularity in the field of computer vision due to its improved accuracy and faster training times. Wang et al. [5] proposed a YOLOv5-based approach for underwater object detection and evaluated their model on a dataset of underwater images. They achieved mAP of 0.81, demonstrating the effectiveness of their approach. [6] proposed an underwater object detection method using TensorFlow. They employed a deep learning approach based on the Faster R-CNN architecture and trained their model on a large-scale underwater dataset. Their approach achieved accurate detection and localization of various underwater objects. The use of TensorFlow facilitated efficient model training and inference, enabling real-time object detection in underwater environments. [7] developed an underwater object detection system using MATLAB. They utilized an ensemble of handcrafted features and classifiers, such as the Histogram of Oriented Gradients (HOG) and Support Vector Machines (SVM), to detect objects in underwater images. Their approach demonstrated good performance in detecting underwater objects, although it relied on manually engineered features and may not scale as effectively as deep learning-based methods.

III. Proposed Methodology

Our research initiates with image classification using convolutional neural networks (CNNs) in the MATLAB interface.CNN is a sort of deep neural network that is used to classify images. CNNs may be used in two ways. One method is to create a CNN from scratch, while another is to use existing ones via transfer learning. It begins by dividing the dataset into training and validation datasets. The ratio of training to validation datasets is 70:30. They are then adjusted to fit the GoogleNet input size. The feature learner layer and output classifier layer are then modified. Finally, the dataset is usedto re-train the revised feature learner layer.
Image classification has been implemented using these 7 simple steps:
1. Dataset loaded
2. Dataset split into training and validation sets with a ratio of 70:30
3. Dataset resized according to the input layer size of the CNN
4. The network inspected; feature learner layer and classification layer identified

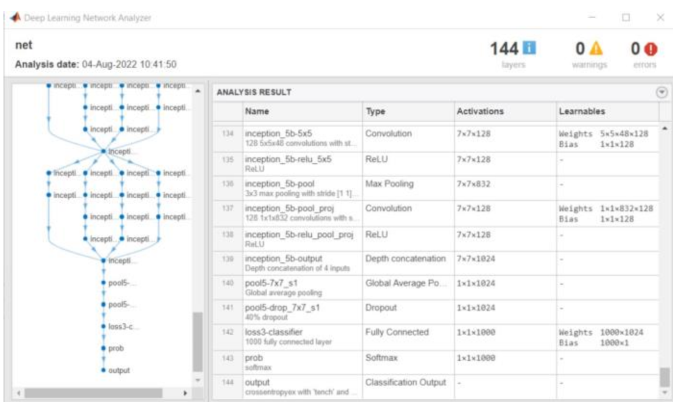

Fig. 1: Depicting the Feature learner and classification layers

5. Feature learner layer and classification layer modified

142	Object Feature Learner 3 fully connected layer	Fully Connected	1×1×3	Weights 3×1024 Bias 3×1
143	prob softmax	Softmax	1×1×3	-
144	Object Classifier crossentropyex	Classification Output	-	-

Fig. 2: Feature learner layer and Classification layer modified

6. Training options defined
7. Network trained
8. Network tested

In the initial stages of our methodology, we conducted experiments using different models, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), ResNet, and GoogLeNet. CNNs, specifically designed for image classification, can be created from scratch or utilized

```
New_Layer_Graph = replaceLayer(Layer_Graph, Feature_Learner.Name, New_Feature_Learner);
New_Layer_Graph = replaceLayer(New_Layer_Graph, Output_Classifier.Name, New_Classifier_Layer);

analyzeNetwork(New_Layer_Graph)

Size_of_Minibatch = 5;
Validation_Frequency = floor(numel(Resized_Training_Image.Files)/Size_of_Minibatch);
Training_Options = trainingOptions('sgdm', ...
    'MiniBatchSize', Size_of_Minibatch, ...
    'MaxEpochs', 6, ...
    'InitialLearnRate', 3e-4, ...
    'Shuffle', 'every-epoch', ...
    'ValidationData', Resized_Validation_Image, ...
    'ValidationFrequency', Validation_Frequency, ...
    'Verbose', false, ...
    'Plots', 'training-progress');
```

Fig. 3: Training options

Fig. 4: Training the network

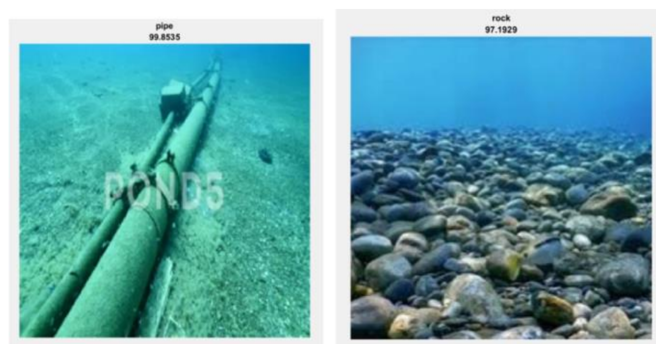

Fig. 5: Testing the network

through transfer learning with pre-trained models. To proceed, our dataset was divided into training and validation sets, with a 70:30 ratio. We resized the images to match the input size of the chosen CNN architecture, and subsequently modified the feature learner layer and output classifier layer accordingly. This involved inspecting the network structure, identifying the relevant layers, and making the necessary modifications. We defined training options, such as learning rate and optimization algorithms, and then proceeded to train the modified network using the training dataset. Ultimately, the network's effectiveness was evaluated using the validation dataset. After exploring these different models, we concluded that YOLOv5 was the most suitable algorithm for our underwater objectdetection task.

The methodology employed in our object detection model encompasses several key steps: Dataset preparation, bounding box annotation, and model implementation. Each of these steps is detailed below.

The initial phase of our project involved the collection and preparation of the dataset, as no suitable underwater object detection dataset was available online. Our dataset consists of a total of 2800 images, comprising both underwater and terrestrial images of rocks and pipes. These images were obtained from various online videos and pictures. For testing purposes, a specific video was used. Next, we proceeded to annotate the training images by drawing bounding boxes around the target objects of interest. We utilized an open-source annotation tool called MakeSense to facilitate this annotation process, as depicted in Figure 1.

The input imagine is separated into an SS grid using YOLO. A single item will be predicted by each grid cell. The subsequent step involved the implementation of the object detection model using the YOLOv5 algorithm, widely recognized as “The world’s friendliest AI architecture” in computer vision. YOLOv5 adopts a grid system approach, dividing an image into grids that individually detect objects contained withinthem. This algorithm stands out among other object detection algorithms due to its simplicity, ease of use, speed, and high accuracy.

To execute the YOLOv5 model, we followed a series of steps: Firstly, we installed the necessary dependencies for YOLOv5. Then, we downloaded a custom YOLOv5 object detection dataset specifically tailored to our project. We proceeded to define the YOLOv5 model configuration and architecture, considering the unique characteristics of our dataset.

Subsequently, we trained a custom YOLOv5 detector using the prepared dataset, allowing the model to learn and adapt to underwater pipe and rock detection. The performance of the YOLOv5 model was evaluated, measuring its accuracy and effectiveness in object detection. The YOLO algorithm employs three key techniques: residual blocks, bounding box regression, and Intersection over Union (IOU). In YOLO, the image is divided into grids, and each grid is responsible for detecting objects within it. Bounding box regression is used to predict the attributes of each bounding box, including width, height, center, and class. By evaluating the amount of overlap between the predicted and actual boxes, IOU is used to enhance the output bounding boxes. By combining these techniques, YOLO achieves efficient and accurate object detection, ensuring that the resultant image contains accurately localized and classified objects.

Finally, we conducted inference using the trained YOLOv5 model on test images, assessing its ability to detect and localize objects accurately. Notably, YOLOv5 exhibits superior speed compared to other algorithms, achieving impressive frame rates of up to 45 FPS. Unlike other approaches such as Faster RCNN that rely on region proposal networks and subsequent recognition, YOLO performs all predictions through a single fully connected layer. This design choice enables YOLO to deliver both increased prediction accuracy and improved intersection over union in bounding boxes, while maintaining its inherent advantage of speed.

$$\mathrm{IOU} = \frac{areaofoverlap}{areaofunion}$$

```
!python train.py --img 640 --batch 2 --epochs 120 --data custom_data.yaml --weights yolov5s.pt --cache
```

Fig. 6: Training YOLOv5s model

```
!python train.py --img 640 --batch 2 --epochs 120 --data custom_data.yaml --weights yolov5s.pt --cache
```

Fig. 7: Training YOLOv5x model

IV. Dataset

Due to the limited availability of publicly available underwater datasets, a custom dataset was created for this project. The dataset was specifically curated to include a diverse range of underwater objects commonly encountered in marine environments, such as rocks, corals, pipes, and various types of fish. To collect the data, we used videos captured using underwater cameras ensuring a variety of lighting conditions, depths, and perspectives. The videos were then preprocessed to extract individual frames, as each frame represents a dis- tinct instance for object detection. Preprocessing steps were involved removing redundant frames and enhancing imagequality. Care was taken to ensure that the dataset captured the complexity and variability of underwater scenes, simulating real-world scenarios as accurately as possible. The dataset wasmeticulously annotated by manually labelling each object of interest within each frame. Each annotated object was assigned a class label corresponding to its category, such as “rock”, “coral” “pipe,” or specific fish species. Bounding boxes were drawn around the objects to indicate their precise locations within the frame.

Fig. 8: Annotating the pipes and rocks in the dataset

the resulting dataset serves as a valuable resource for training and evaluating the underwater object detection model based on YOLOv5. Its composition of diverse underwater objects and realistic environmental conditions enables the model to learn and generalize effectively, facilitating accurate detection and classification of objects in real-world underwater scenarios.

V. Results

The YOLO algorithm has proven to be a powerful tool for underwater object detection. Throughout our research, we have observed the performance, accuracy, and speed variations among different versions of YOLO, which are crucial factors to consider when selecting an appropriate model for specific applications.

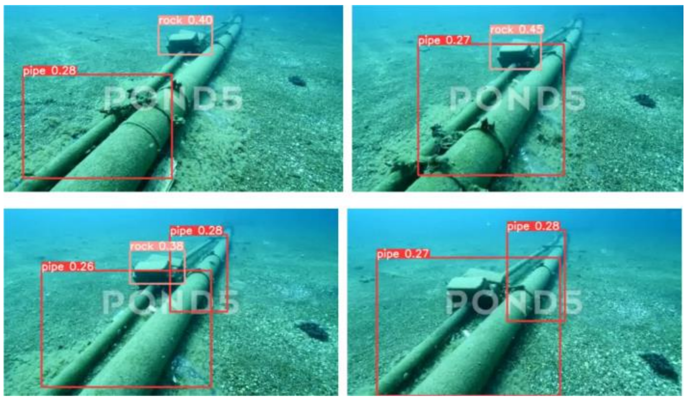

Fig. 9: Frames from the predictions made in the output video

YOLOv3 strikes a balance between accuracy and speed, making it suitable for real-time underwater object detection tasks. However, subsequent iterations have demonstrated notable improvements in performance. YOLOv4, with its innovative features such as the CSPDarknet53 backbone and PANet, exhibits enhanced accuracy, particularly for small object detection, albeit with increased computational requirements. On the other hand, YOLOv5 has garnered attention for its impressive accuracy, reduced model size, and faster training times, making it an appealing option for resource- constrained underwater environments. Comparisons between these versions have shown that YOLOv4 generally out per- forms YOLOv3 in terms of accuracy, while YOLOv5 achieves comparable accuracy to YOLOv4 while being more computationally efficient. The choice of YOLO version should be based on the specific requirements of the underwater object detection task, considering the trade-offs between accuracy and speed. By leveraging the advancements in different YOLO versions, researchers and practitioners can tailor their choice to best suit their needs and constraints, ultimately enabling more effective and efficient underwater object detection systems.

Conclusion and Future Scope

In conclusion, our study focused on underwater object detection using the YOLOv5 algorithm, addressing the challenges posed by underwater environments. Through the development and evaluation of our custom dataset and the implementation of YOLOv5, we successfully demonstrated the effectiveness of the algorithm in detecting and localizing various underwater objects, including rocks, pipes, and fish. However, there are still areas where we can improve. We should find ways to improve the detection of small and hidden objects underwater, which can be challenging. Using different techniques to enhance the images and combining different types of data, like visuals and sounds, could help us improve the accuracy of object detection underwater.

Acknowledgment

We would like to extend our gratitude to Dr. Shruthi M L J, Associate Professor, PES University, Bangalore, India for her support and guidance throughout this project. We also extend our thanks to PES University for providing us with the necessary resources and facilities to conduct this research.

References

[1] Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion. Remote Sens. 2021, 13, 4706.

[2] Li, Z., Li, Y., Li, Y. (2019). Underwater object detection based on improved YOLOv3. IEEE Access, 7, 143225-143235.

[3] Liu, J., Wang, Z., Liu, B., Zhang, J., and Luo, H. (2020). Lightweight object detection model for underwater images. IEEE Access, 8, 130317-130326.

[4] Zhang, Y., Zeng, Y., and Cheng, S. (2020). Underwater object detection based on improved YOLOv3 algorithm. Journal of Ambient Intelligence and Humanized Computing, 11(4), 1509-1520.

[5] Wang, Q., Chen, W., Li, C., Huang, S., and Li, B. (2021). Underwater object detection based on YOLOv5. IEEE Access, 9, 10468-10477.

[6] Y. Wang, J. Liu, S. Yu, K. Wang, Z. Han and Y. Tang, ”Underwater Object Detection based on YOLO-v3 network,” 2021 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 2021, pp. 571-575, doi: 10.1109/ICUS52573.2021.9641489.

[7] T. Leisheng, X. Hongli, W. Han, T. Dongxu and G. Lei, ”Research on Collaborative Object Detection and Recognition of Autonomous Underwater Vehicle Based on YOLO Algorithm,” 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 1664-1669, doi: 10.1109/CCDC52312.2021.9601610.

[8] H. Wang et al., ”A YOLOv5 Baseline for Underwater Object Detection,” OCEANS 2021: San Diego – Porto, San Diego, CA, USA, 2021, pp. 1-4, doi: 10.23919/OCEANS44145.2021.9705896.

[9] R. Jaffari, M. A. Hashmani, C. Carlos Reyes-Aldasoro, N. Aziz and S. S. Hussain Rizvi, ”Deep Learning Object Detection Techniques for Thin Objects in Computer Vision: An Experimental Investigation,” 2021 7th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 2021, pp. 295-302, doi: 10.1109/ICCAR52225.2021.9463487.