Journal of Scientific & Industrial Research
Vol. 81, October 2022, pp. 1050-1060
DOI: 10.56042/jsir.v81i10.61397

Enhancement and Detection of Objects in Underwater Images using Image Super-resolution and Effective Object Detection Model

R Arumuga Arun*, S Umamaheswari, B Nafesha, V Makesh Arvindan & Vengam Udaya Kumar
Anna University-MIT Campus, Chennai 600 044, Tamil Nadu, India

Received 12 March 2022; revised 17 September 2022; accepted 17 September 2022

It is imperative to build an automatic underwater object recognition system in place to reduce the costs of underwater inspections as well as the associated risks. An effective method of detecting underwater objects from underwater images of aquatic after enhancing them using the Image Super-resolution technique is proposed in this study. The proposed approach comprises of two major sections, Underwater Image Enhancement, and Object detection. To enhance the underwater images, a lightweight Reduced Cascading Residual Network (RCARN) is proposed that imposes the Image Super-resolution technique. Later, the enhanced images generated by the RCARN model are supplied for the object detection process, where a significant object detection model, YOLOv3 is employed in this study. To improve its performance, this YOLOv3 is trained on one of the largest datasets, the COCO data, followed by being fine-tuned using enhanced Underwater images. The dataset utilized in this work contains 6 classes of underwater objects namely dolphin, jellyfish, octopus, seahorse, starfish, and turtle. All these images are actual real field images collected from various sources. With this proposed approach, a better overall ACS and mAP of 95.44% and 75.33% are achieved here, which are improved by ~8.75% and ~15%, respectively when compared to actual collected low-resolution images.

Keywords: CNN, Computer vision, Deep learning, Image enhancement, Object detection, Underwater objects

Introduction

As you get closer to a particular depth, underwater images degrade in contrast, become blurry, and suffer from color distortion. The restoration and enhancement of underwater images have therefore become challenging.[1–3] Light propagating through water is absorbed and reflected, which influences underwater imaging. As an example, the internal optical property (IOP) of water measures the amount of light that the water absorbs. Depending on their wavelengths, the colors of light diminish as water depth increases.

A high-resolution image has a higher pixel density and contains more information about the original scene. In addition, in computer vision applications, high resolution is prevalent in pattern recognition and image analysis for better performance. Because of the blurring, poor contrast, and uneven illumination that affect underwater images, computer vision applications find it challenging to classify objects. To overcome the problems, enhancing the quality of the input image is done by using Super Resolution Technique. Through a technique known as super-resolution (SR), low-resolution images are merged to create high-resolution images.[4,5]

Guo *et al.* suggested a multiscale dense Generative Adversarial Network (GAN) for enhancing the underwater image through a mapping of non-distorted images to distorted ones using a nonlinear method.[6] The core component of this proposed generator was a multiscale dense block that facilitates better utilization of feature maps for improvising the quality of enhanced images, which was the inspiration for our work. Using deep learning techniques, Yeh *et al.* developed a method for removing haze from images, which combines Multi-Scale Residual Learning (MSRL) with image decomposition.[7] The key idea gathered from this was utilizing multiscale residual connection on a U-Net model for mapping the hazy and haze-free base components, resulting in that haze from images being removed.

The Soft Edge assisted Network (SeaNet) developed by Fang *et al.* is designed for reconstructing images with high-quality SR with the help of the image soft-edges.[8] The key idea of this proposed work was that, instead of increasing the models' depth, the authors integrated the images' prior knowledge into the model. The limitation of this approach was that, it requires a feature engineering

*Author for Correspondence
E-mail: arun6f.rajesh@gmail.com

process, which is more challenging with low-resolution underwater images.

In deep learning, a model learns explicitly from text, image, sound, or images to perform tasks and can achieve incredible accuracy, sometimes more than performance at the human level.[9] The main benefits of using deep learning are that they allow feature engineering to be done on its own as well as helps to improve performance.[10,11] Observable faults that are difficult to train can be detected with the aid of deep learning, such as minimal product labeling errors, etc. It is hard to interpret unstructured data for most machine learning algorithms, which means it remains less used, and this is actually where deep learning is effective.

However, for image recognition[12], detection and localization[13], segmentation, classification[14], and so on, a neural network of one or more convolutional layers is used that is termed as Convolutional Neural Network (CNN).[15] They are composed of neurons, where each neuron has a weight and bias that can be trained. Predicting a single label requires a classification task. Various real-time tasks require more than one class label to be predicted. Therefore, this suggests that it is not mutually exclusive to class names or class membership. These tasks are defined as multiple-label classification or, for short, multi-label classification,[16] This task is also called an object detection task.

The process of detecting objects is a crucial task in computer vision. Object detection tells us the precise position of objects in an image, while image classification identifies what the image is. The pathway to achieve this involves training an encoder to produce a bounding box and associated class probabilities for each object in an image. YOLO is one of the significant deep learning-based object detection models, that can identify objects by using only one look. It is a regression-based object detection approach, which divides every image into multiple grid cells to detect objects, where every grid cell generates numerous bounding boxes, their confidence values, and class probabilities. YOLO is successful because it has a high degree of accuracy and also can run in real-time.[17,18]

Malhotra *et al.* illustrated a comparison among the well-renowned approaches of R-CNN, Fast R-CNN, and YOLO to achieve object detection with its architectures.[19] Real-time detections cannot be accomplished by either RCNN or Fast RCNN, while real-time classification can be achieved by YOLO with reasonable speed. In this study, the images in the dataset are directly used without any image enhancement. Fang *et al.* developed a Tiny-YOLOv3 model to reduce the computational complexity of the actual YOLO model.[20] Based on its experimental results; Tinier-YOLO is more efficient than tiny YOLOv3 but performs worse than MobileNet SSD. The image enhancement process was not employed in this study prior to the model's training since the training data were of sufficient clarity.

The efficient method of attaining the models' significant performance in a shorter training period with lesser training is the transfer learning technique. The key idea of work of the authors Garcia-Dominguez *et al.* was to construct an accurate CNN-based classification model on smaller instances dataset with the transfer learning technique.[21] Huo *et al*. proposed a multi-class classification model for searching and rescuing drowning victims.[22] The authors improvised their proposed model's accuracy to 97.76% by employing the pre-trained CNN layers in their model.

Materials and Methods

The primary objective of the proposed approach is to build a computer vision system using a deep learning-based object detection model to classify and detect the objects in underwater images accurately. The workflow of this proposed approach is visualized in Fig. 1. Under the pre-processing section, initially, the images in the dataset are brought into the augmentation process to increase the dataset

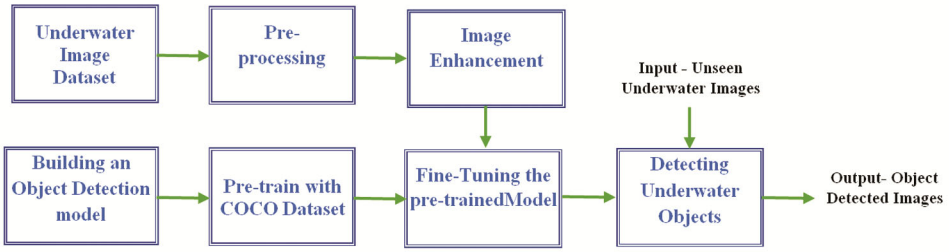

Fig. 1 — The work-flow of the proposed Under Water Objects Detection approach

instances. Later, these augmented images are applied for the image enhancement process to resolve the problem associated with the low resolution of underwater images. The underlying deep learning model for detecting the objects is trained initially using one of the larger datasets, the COCO dataset, followed by fine-tuning using these enhanced images. As a final step, the trained Object Detection model is utilized for detecting the objects in the unseen underwater images.

Data Preprocessing

A dataset is considered the fuel for constructing a successful Deep Learning (DL) model. The dataset utilized in this work contains six classes, namely, dolphin, jellyfish, octopus, seahorse, starfish, and turtle. The instances for these six classes are collected from various sources. Building an accurate DL model always demands huge data/images since feature engineering is done on its own. But, in many cases, it is hard to make a huge instance dataset, and the available data is also not sufficient to obtain a good performing DL model. Similarly, in this case, collected images are not sufficient to make a successful model, and further, the number of instances among the classes is imbalanced.

Initially, every image of this dataset has been resized to 256 × 256 pixels and then taken into the augmentation process to create more instances, finally making it more balanced among the classes in the dataset. As part of this augmentation process, multiple augmentation operations like horizontal and vertical flipping, width and height shifting, rotation, and zooming are applied to the available images to create the transformed versions of them.[23] With this transformed version of images, the dataset becomes a rich and sufficient one with many different instances for building a better-performing model.[24] It also helps to avoid overfitting and facilitates building a model in a generalized manner. In Table 1, the details of the dataset before and after augmentation are presented.

Table 1 — The details of the dataset

Class No	Class Name	Before Augmentation	After Augmentation	Data Split after Augmentation	
				Train	Test
C01	Dolphin	120	450	360	90
C02	Jellyfish	150	460	368	92
C03	Octopus	115	460	368	92
C04	Seahorse	130	460	368	92
C05	Starfish	100	440	352	88
C06	Turtle	118	445	356	89
	Total	733	2715	2172	543

The majority of the underwater images acquired are low-quality in nature, with only a few of them being of decent resolution. Underwater objects cannot be detected and classified accurately while using these low-resolution images for the object detection task. Here are a few examples of low-resolution images shown in Fig. 2. Hence, these low-resolution underwater images are passed to the proposed image enhancement process.

With the hold-out strategy, these enhanced images are sliced into an 8:2 ratio for testing and training.[25] The annotation process is then applied on these training and testing datasets. Bounding boxes are attached to the objects in the underwater images, called annotations. The label file contains details such as object class and bounding box coordinates are stored. This annotation process was done using LabelImg tool. The label file format is text since YOLOv3 supports text label files. In a label file, there are five fields, in which the first field represents a class number, the second and third fields represent the x, y coordinates of an object's center point, and the fourth and fifth fields represent the width and height of the bounded object. The annotation process, a sample label file, and its format are visualized in Fig. 3. Finally, these

Fig. 2 — Few of the sample low-resolution underwater images

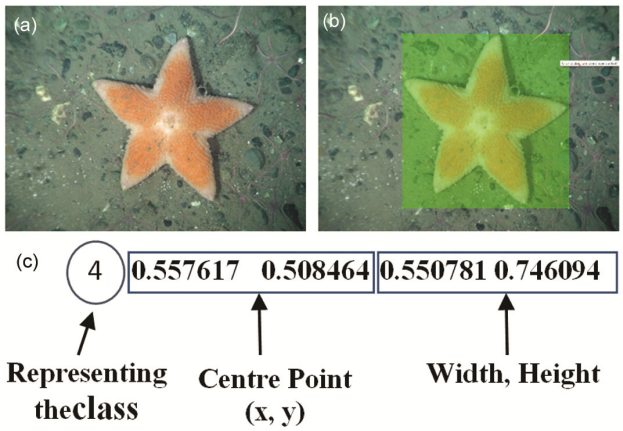

Fig. 3 — The Annotation process, a) Original file, b) Bounding the underwater object, c) Corresponding label file in text format

enhanced images are used to detect underwater objects through the object detection process.

Image Enhancement

Low-resolution images are enhanced using the image enhancement process. A typical workflow of the image enhancement process is depicted in Fig. 4.

Building RCARN Model

Reduced Cascading Residual Network (RCARN)

A deep learning model called Reduced Cascading Residual Network (RCARN) is proposed for upgrading underwater images by improving the resolution of poor-resolution underwater images. It contains a total of 28 convolution layers and each convolution operation uses a stride value of 1 and the same-padding convolution. Among these 28 convolution layers, 12 layers use a 1 × 1 sized kernel and the remaining 16 layers use a 3 × 3 sized kernel. In this, each 3 × 3 and 1 × 1 convolution layer is followed by a Batch Normalization (BN) Layer, in addition to that, every 1 × 1 convolution layer also has a ReLU layer after the BN layer. An illustration of the Reduced Cascading Residual Network (RCARN) is shown in Fig. 5 which is a modified CARN architecture developed in the work.[26]

The proposed RCARN model is organized as three cascading blocks, three concatenation units, and seven convolution layers, of which four are 3 × 3 and three 1 × 1 layers. Between a 1 × 1 and a 3 × 3 convolution layer is a cascading block and concatenation unit pair. An illustration of the proposed RCARN architecture's configuration is mentioned in Table 2.

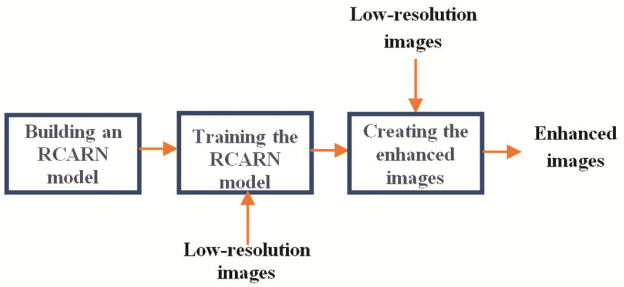

Fig. 4 — The work-flow of the proposed Image Enhancement Process

The role of the concatenation unit is to combine the feature maps that help improve the performance of a deep learning model.[27] An important reason for employing the concatenation unit here is to concatenate feature maps that precede and follow a cascading block. It facilitates the improvisation of the RCARN model's performance. But, on the other side, it increases the number of parameters generated by a model. To prevent this problem, the RCARN model adds a 1 × 1 convolution layer after each concatenation unit. It aids in the reduction of the feature map's depth without compromising the model's performance. As a result, the model's computational complexity is reduced.

Cascading Block

Cascading block is a core functional unit of the proposed RCARN model that differentiates this RCARN model from the CARN model presented in the work.[28] The CARN model's cascading block featured global-level cascading connections, allowing each convolution layer's output to be passed to every next level layers.[28] Even though it enhances better utilization of feature maps it increases the computational complexities.[4] To address this, we proposed a modified cascade block that generates fewer computational parameters without compromising the model's efficiency.

Table 2 — The architectural configuration of the proposed RCARN model

S. No	Layer / Unit	Kernel Size	No of Filters/ Units
1	CONV+ BN	3 × 3	32
2	Cascading Block	3 × 3, 1 × 1	1
3	Concatenation Unit	—	1
4	CONV + ReLU + BN	1 × 1	64
5	CONV + BN	3 × 3	32
6	Cascading Block	3 × 3, 1 × 1	1
7	Concatenation Unit	—	1
8	CONV + ReLU + BN	1 × 1	64
9	CONV + BN	3 × 3	32
10	Cascading Block	3 × 3, 1 × 1	1
11	Concatenation Unit	—	1
12	CONV + ReLU + BN	1 × 1	64
13	CONV + BN	3 × 3	32

CONV – Convolution Layer,
BN – Batch Normalization Layer,
ReLU – ReLU Activation Layer

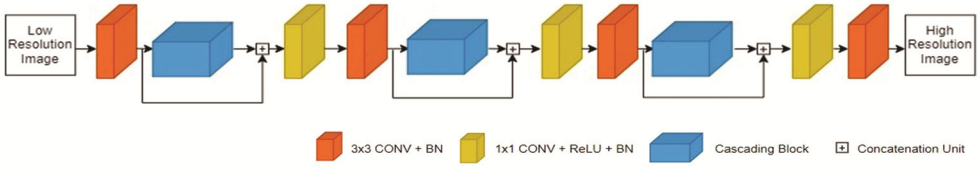

Fig. 5 — The structure of the proposed Reduced Cascading Residual Network (RCARN) model

The proposed cascading block is built using seven convolution layers and three concatenation units, which include three 1 × 1 convolution layers, and four 3 × 3 convolution layers. In contrast to the cascading blocks of the CARN model, the proposed cascading blocks have three concatenation units that create cascading connections rather than global cascading connections. It enhances the better utilization of feature maps within the model.[29] In addition, after every concatenation unit, the point convolution layer is employed to optimize the computational parameters. The structure of the proposed cascading block used in this RCARN model is visualized in Fig. 6.

Furthermore, the suggested cascade block can be formed in four different ways by varying the number of convolution filters used in the 3 × 3 convolution layers. In Table 3 the detailed model configuration of these four versions is shown. The proposed RCARN model is developed in four varieties by these cascade block variants: RCARN 32-32, RCARN 32-48, RCARN 32-64, and RCARN 64-64.

Training the RCARN Model

In the beginning, all the proposed RCARN model variants are trained on the training dataset in Table 1 for 200 epochs. The resulting trained RCARN models are capable of generating enhanced underwater images from low-resolution images. Later, the efficiency of these trained RCARN model variants is evaluated using the test dataset images.

Creating the Enhanced Underwater Images

Once the RCARN model is constructed and trained successfully, each low-resolution image in the dataset is fed into the model. It transforms all the low-resolution underwater images into enhanced underwater images with the Super Resolution technique. Later, all these enhanced underwater applied to the object detection process for detecting underwater objects. As a result of these improved images, objects can be detected with higher accuracy than with the original low-resolution ones. The proposed image enhancement algorithm is mentioned below.

Algorithm: Image Enhancement Algorithm

Input: Low Resolution Image

Output: Enhanced Image

1. For every image (low resolution and high resolution) from the dataset.
    1.1 For every img from the file:
        1.1.1 Load the image in YCbCr color format.
        1.1.2 Resizing the image to 256 × 256
    1.2 End For
2. End For
3. Build the model as per the configuration mentioned in Table 2.
4. Compile the proposed model along with an Adam Optimizer (learning rate = 0.001).
5.Train and Evaluate the model using fit() and evaluate() function.
6. Load the trained model in the .h5 file format.
7. Initialize the new_img parameter as 0.

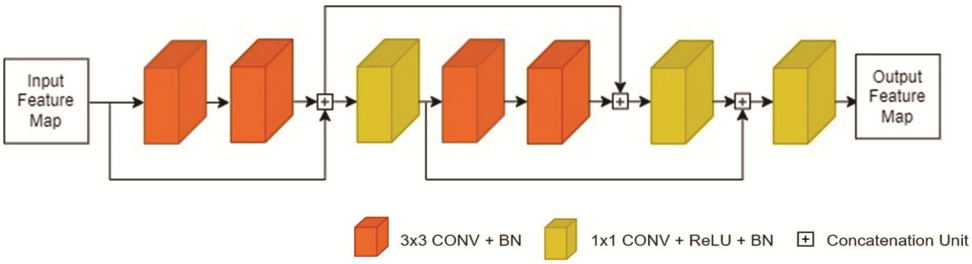

Fig. 6 — The structure of the proposed Cascading Block used in the proposed RCARN model

Table 3 — Architectural configuration of Proposed Cascading Block

S. No	Layer	Kernel Size	No of Units/ Filters			
			32-32	32-48	32-64	64-64
1	Conv + BN	3 × 3	32	32	32	64
2	Conv + BN	3 × 3	32	48	64	64
3	Concatenation Unit	—	1	1	1	1
4	Conv + BN + ReLU	1 × 1	64	64	64	64
5	Conv + BN	3 × 3	32	32	32	64
6	Conv +BN	3 × 3	32	48	64	64
7	Concatenation Unit	—	1	1	1	1
8	Conv + BN + ReLU	1 × 1	64	64	64	64
9	Concatenation Unit	—	1	1	1	1
10	Conv +BN + ReLU	1 × 1	64	64	64	64

Conv - Convolution layer, BN – Batch Normalization Layer, ReLU – ReLU Activation Layer

8. For each image as new_img from the test dataset:
   8.1 Reshape the new_img and generate the scaled new_img using scaled_new_img = resized_new_img/255.0
   8.2 Sharp the generated new_img into the size of 256 × 256
   8.3 Load the sharpened image as an enhanced image into the output file.
   8.4 Increment value of image by 1.
9. End For

Detection of Underwater Objects

Building an Object Detection Model

For detecting objects from enhanced underwater images, YOLOv3 has been utilized in this work. YOLOv3 is one of the significant object detection models used in many applications which follow a one-stage detection algorithm.[30] This model is implemented based on the DarkNet framework. This YOLOv3 model employs modified DarkNet as a backbone network, which comprises 106 convolution layers, and its original configuration is 53 layers in its previous versions. An architecture diagram of the YOLOv3 is illustrated in Fig. 7.

The salient feature of the YOLOv3 model is a multi-scale detector since it detects objects at three different scales. This YOLOv3 model is a fully convolutional network that generates output by applying a 1 × 1 kernel to a feature map. In this, feature maps of three different scales, such as down-sample the input image by 32, 16, and 8 are used to detect objects using 1 × 1 kernels at three places, such as 82nd, 94th, and 106th layers within the network.

Pre-training the Object Detection Model

In general, the process of building deep learning-based object detection models always demands a huge training data and more training time. Transfer learning is an excellent strategy for building an accurate object detection model even with a short training period and a small number of training images.[31,32] Because, transfer learning involves pre-training the CNN model on large datasets such as ImageNet[33], Pascal[34] and COCO[35] dataset, and later fine-tuning the model weights is done based on the target dataset.This study takes advantage of transfer learning by training the YOLOv3 model with the COCO dataset, and later models' trained weights are fine-tuned using actual underwater images.

Fine-tuning the Pre-Trained Model

As opposed to training the model from scratch, the pre-trained weights will be fine-tuned to fit the actual dataset rather than being learned from scratch. As a result, using transfer learning, developing an effective model with a smaller dataset and a faster training period becomes achievable. As a result, it's useful for this underwater object detection work wherein collecting more training images is challenging.

While feeding these low-resolution images to the object detection task, it is hard to identify the underwater objects precisely. Hence, all these low-resolution images are fed through the image enhancement process, in which the proposed RCARN model enhances all the low-resolution images into high-resolution images using the Super-Resolution technique. These enhanced underwater images are used to fine-tune the YOLOv3 model for detecting underwater objects. Later on, this fine-tuned YOLOv3 model is applied to the object detection process, such that it can detect underwater objects from unseen underwater images.

Experimental Environment and Evaluation Metrics

Design of Experimental Environment

The proposed RCARN and YOLOv3 models are developed on Colab-Pro, a platform that features GPU

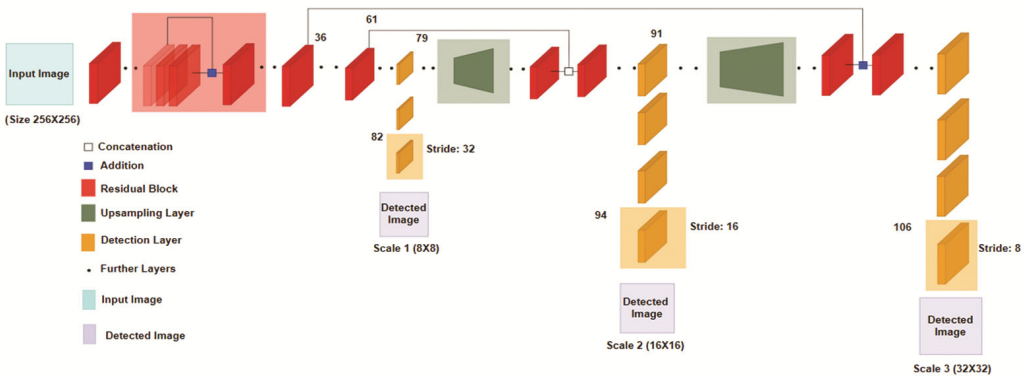

Fig. 7 — The structure of the YOLOv3 object detection model utilized in this Underwater Object Detection task

K8, T4 and P100, virtual RAM of 25 GB, and virtual memory space of 166 GB. This study used 2715 underwater images of 6 classes of underwater objects after augmentation, as outlined in Table 1. By using the holdout method, training and testing datasets are created from this dataset in an 8 to 2 ratio.[25] Hence, the training and testing dataset contains 2172 images and 543 images, respectively. The reason behind using the hold-out method is it makes the model more effective on unseen images.[36]

As part of the image enhancement process, all four RCARN variants, namely RCARN-32_32, RCARN_32-48, RCARN_32-64, and RCARN_64-64, are trained from scratch for 200 epochs using the training dataset mentioned in Table 1. The batch_size used here is 8 images/batch, so every training epoch contains 272 steps, and its learning rate is 0.001. In this image enhancement process, a categorical_cross-entropy loss function and the ADAM optimizer are used.

During the object detection process, a location for each object will be determined. All the images in the dataset must be annotated to accomplish this. With the annotation tool LabelImg, the annotation task is accomplished.As a preliminary step, the underlying YOLOv3 dataset used here is pre-trained on COCO and fine-tuned afterward on the target underwater images dataset. Similar to the image enhancement process, the same training and testing datasets split is used here. While feeding the training images to this YOLOv3 model for fine-tuning, the training images pass via the trained RCARN model that enhances all the underwater images before it reaches the YOLOv3 model.

This fine-tuning process is carried out for 8000 steps, and the batch size used here is 16 images/batch. In this object detection task, categorical_cross_entropy is used, which measures the loss between the actual label and the predicted one. Further, the loss between prediction and actual label is optimized by incorporating the ADAM optimizer with learning_rate = 0.001.

Evaluation Metrics

The performance of the proposed RCARN model and object detection model is evaluated using the following metrics

1. Peak Signal to Noise Ratio (PSNR)
2. Structural Similarity Index Measure (SSIM)
3. Average Confidence Score (ACS)
4. mean Average Precision (mAP)
5. Total Number of Parameters (TNP)

Peak Signal to Noise Ratio (PSNR)

Peak signal-to-noise ratio (PSNR) is the term of the performance metrics for the ratio of the highest possible power of the signal to the power of the corruptive noise that degrades the representation fidelity. In the case of signals with a wide dynamic range, the PSNR is computed using the logarithmic decibel scale. The mathematical form PSNR is mentioned in Eq.1

$$PSNR = 10\log_{10}\frac{(peak\ value)^2}{MSE} \quad \ldots (1)$$

$$MSE = \frac{1}{mn} * \sum_0^{m-1}\sum_0^{n-1}||u(i,j) - v(i,j)||^2 \quad \ldots (2)$$

where, peak value is either the value described by the user or the value that was selected from the image data type's range, and the MSE represents the mean square error computed from the reference image and the user's original image, u and v are the array value of the original image and of the degraded image, respectively, m and n represent the number of rows and columns of pixels of the images, respectively, and i and j represent the index of each row and column, respectively.

Structural Similarity Index Measure (SSIM)

Structural Similarity Index Measure (SSIM) is a performance metric that measures the degree of similarity between two images. The SSIM index is used for calculating the consistency of an image by comparing it to a reference image and initial image. The mathematical form SSIM is shown in Eq. 3.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\,\sigma_{xy} + c_2)}{(\mu_x^1 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad \ldots (3)$$

$$c_1 = (k_1 L)^2 \quad \ldots (4)$$

$$c_2 = (k_2 L)^2 \quad \ldots (5)$$

where, $\mu_x$and μ_y represent the average of image x and image y respectively. σ_x^2 and σ_y^2 represent the variance of image x and image y respectively. σ_{xy} the covariance of images x and y. c_1 and c_2 represent the two variables that are used to stabilize the division with weak denominators. L represents the dynamic range of the pixel-values, typically this is a power of (2, (bits per pixel - 1)), and the default values of k_1 and k_2 are −0.01 and −0.03.

Average Confidence Score (ACS)

In object detection tasks, the confidence score is an evaluation metric that reflects how confident the model is that its predicted bounding box has an object inside. Also, how accurate is the bounding box that

the model predicts. The average confidence score (ACS) is the mean of the confidence score predicted by the model for the images in the test dataset. Here, ACS is calculated for each object class. The mathematical form of ACS is shown in Eq. 6.

$$ACS = \sum_{i=1}^{n} CS_i \quad \ldots (6)$$

$$CS = IoU \times CP \quad \ldots (7)$$

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad \ldots (8)$$

where, CS represents Confidence Score, n represents the total number of images in the dataset, IoU represents Intersection over Union, CP represents Class Probability which is the probability of the class present in the box.

mean Average Precision (mAP)

The mean Average Precision (mAP) is an average of the Average Precision (APr) values, where the Average Precision (APr) is calculated for every class. The mathematical form of mAP is given in Eq. (9).

$$mAP = \frac{\sum_{k=1}^{M} A\Pr_k}{M} \quad \ldots (9)$$

where, mAP represents the mean Average Precision, APr_k represents k^{th} class Average Precision value, and M represents the number of classes in total.

Total Number of Parameters (TNP)

The total number of parameters (TNP) is a summation number of parameters generated by each layer in the model. The mathematical form of TNP is shown in Eq. 10.

$$TNP = \sum_{i=1}^{n} NPi \quad \ldots (10)$$

$$NP = KS \times KS \times NIC \times NOC \quad \ldots (11)$$

where, NP refers to the no. of parameters generated by a convolution layer, the number of layers is represented by n, KS represents the kernel size. The parameters NIC and NOC represent the number of input and output channels, respectively.

Results and Discussion

The proposed approach includes two major parts: Underwater Image Enhancement and Underwater Object Detection. The first section of this work implements four RCARN model variants to enhance underwater images of low resolution. In the second section, the YOLOv3 model is utilized for detecting underwater objects from the enhanced underwater images generated by the RCARN model.

In the first section, based on Ahn *et al*. (2018)[28], the CARN model is implemented for this study to compare the efficiency of the proposed model variants and determine the most efficient one. The training process on all the RCARN models' variants accomplished by the training dataset and its details is mentioned in Table 1. The performances of these models are assessed by using the metrics, namely PSNR and SSIM. The corresponding results and the parameters generated by each model are recorded in Table 4.

All the proposed models outperformed well when compared to the existing CARN model (Table 4), and its TNP values are also lesser than the existing CARN model, except for Proposed RCARN_64-64. In comparison to other RCARN variants and the existing CARN model, the proposed RCARN_32-32 has less TNP value, which is 0.2 Million. Usually, the value of metrics PSNR and SSIM will be high if a better resolution enhancement is achieved in the images.

Among these proposed RCARN variants, except the RCARN_32-64 model, the SSIM value decreases with the increase in PSNR value. In the model RCARN_32-64, the increase in PSNR value does not affect the SSIM value. Further, this proposed RCARN_32-64 model also achieved the top PSNR and SSIM values, such as 30.59 and 0.908, when compared to others. But its TNP value is 0.37 Million, which is a little higher, however, the difference is very small. The proposed RCARN_32-64 model serves the best results regarding image enhancement. Hence, this proposed RCARN_32-64 model is used to enhance the underwater images, and these generated enhanced images are utilized for training the object detection model.

The detection of underwater objects using the YOLOv3 model is assessed using the ACS and mAP metrics. This ACS is calculated on every class basis of the six different classes of objects used in this work. The confidence Score, commonly called the objectness score, measures the probability of an

Table 4 — Comparison of the proposed metrics of RCARN variants with the existing model

S. No	Model Variants	NoP *(in Million)*	PSNR	SSIM
1.	CARN[27]	0.62	26.58	0.85
2	RCARN_32-32	0.20	29.59	0.90
3	RCARN_32-48	0.27	29.83	0.87
4	RCARN_32-64	0.37	30.59	0.90
5	RCARN_64-64	1.38	30.07	0.86

object being present in the predicted location. The ACS is the average of all the confidence scores predicted by the model, here calculated for the images in the test dataset. The learning curves of the ACS and the IoU values over the fine-tuning steps are visualized in Fig. 8.

Further, to understand how the image enhancement process influences the object detection process, the ACS and mAP are calculated for both the actual low-resolution images and the enhanced images produced by the RCARN_32-64 model. An analysis of the ACS metric values attained is presented in Table 5, while an analysis of the ACS metric values actual versus enhanced images is shown in Fig. 9.

As seen in Table 5, the average improvement in overall ACS for the YOLOv3 model on enhanced images was~8.5% compared with the overall ACS value for the actual image. A few detected images of the enhanced version are shown in Fig. 10. Hence, the proposed RCARN_32-64 model helps the YOLOv3 model for detecting underwater objects more precisely by enhancing the low-resolution underwater images.

The most common metric for assessing an object detection model's performance is mAP. Here, the mAP is computed for both actual image low-resolution and enhanced images at two ranges of IoU = 0.75 and IoU = 0.5: 0.9. From Table 6, at IoU = 0.5 level, the attained mAP value of both actual low-resolution and enhanced images are considerably good, 85.05 and 90.5, respectively. Compared to actual images, enhanced images offer an improvement

Table 5 — The Comparison ACS values achieved for Actual and Enhanced images

Class Name	Achieved ACS (%)		
	Actual image	Enhanced image	Improvement
Dolphin	84.01	94.51	10.5
Jellyfish	85.15	93.54	7.99
Octopus	86.71	96.02	9.31
Seahorse	87.83	94.83	7.00
Starfish	88.53	97.95	9.42
Turtle	88.12	96.44	8.32

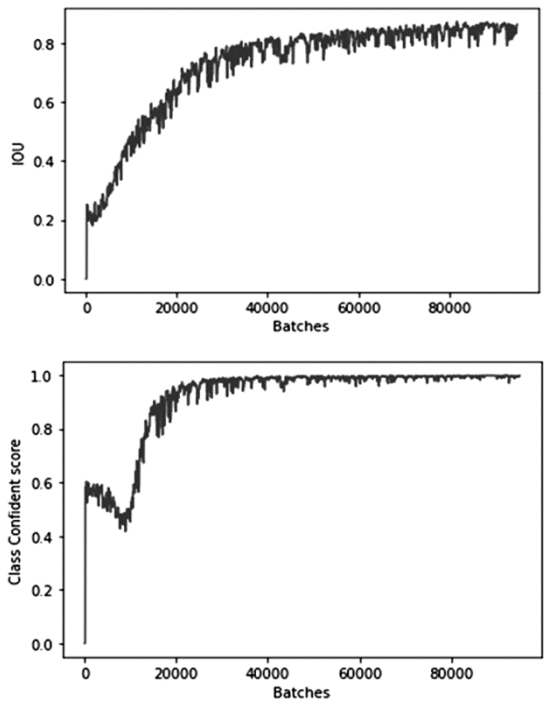

Fig. 8 — The learning curves of IOU and Confidence score over fine-tuning steps, a) Intersection over Union (IoU) learning curves b) Confidence Score learning curves

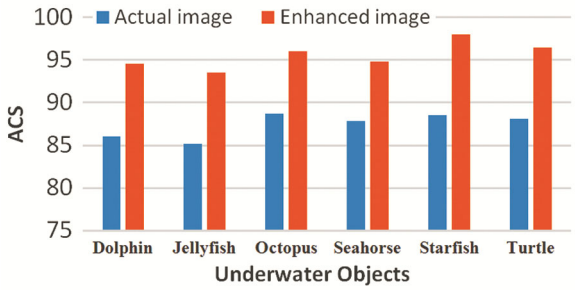

Fig. 9 — The comparison of ACS values for actual images vs enhanced images

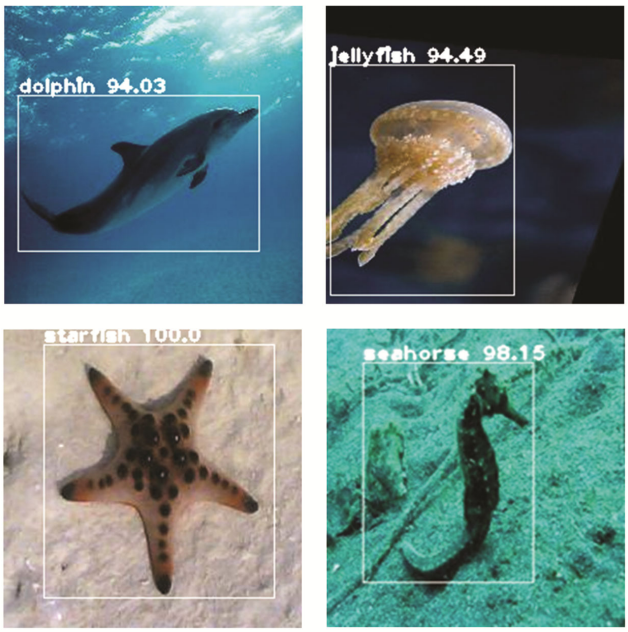

Fig. 10 — Few of the sample deselected enhanced underwater images

Table 6 — Comparison mAP achieved by YOLOv3 with actual and enhanced images

S. No	Model	Training and Testing Images Type	mAP (%)	
			@ IoU = 0.5	@ IoU = 0.5:0.9
1	YOLOv3	Actual low-resolution images	85.05	60.51
2	YOLOv3	Generated enhanced images	90.25	75.33

on mAP of ~5%. But, at IoU = 0.5:0.9 level, the mAP value of the actual images is very low, is 60.51 in contrast to the images with the enhancement process by the RCARN model, the mAP is increased to 75.35% and the improvement in mAP is ~15%. From Tables 5 & 6, we demonstrate that the image enhancement process using the RCARN model helps to improvise the detection capability of the YOLOv3 model in the underwater objects detection process in this study.

Conclusions

In this study, an effective method is presented to detect underwater objects that employ image enhancement using the Image Super-resolution technique before detecting the objects. To perform an underwater image enhancement and object detection, a proposed RCARN_32-64 and a significant YOLOv3 object detection model are employed in this study. The RCARN_32-64 model is constructed in a way of enhancing the images with better PSNR and SSIM values with lesser computational complexity. Further, the performance of the object detection is improvised by Transfer Learning technique, where the YOLOv3 model is pre-trained using COCO data and later fine-tuned using enhanced underwater images. With this proposed approach, the ACS and mAP are improved by ~8.75 % and ~ 15%, respectively. It helps to build an autonomous aquatic inspection system that restrains the expenses and risks associated with the aquatic inspection process.

In this study, we have utilized six underwater species. As part of our future work, we shall be planning to build a model to recognize more underwater species which helps to survey the underwater species in a particular aquatic region. Further, we shall also be planning to build a model that enhances the low-resolution video to high-quality video on the fly.

References

1 Yang M, Hu J, Li C, Rohde G, Du Y & Hu K, An in-depth survey of underwater image enhancement and restoration, *IEEE Access*, **7** (2019) 123638–123657.

2 Er M J & Jie C, Research challenges, recent advances and benchmark datasets in deep-learning-based underwater marine object detection: A review (2022), TechRxiv. Preprint, Availble: https://doi.org/10.36227/techrxiv.19350389.v2.

3 Jagatheswari S & Viswanathan R, Image magnification and demagnification using fuzzy lattice morphological transformation, *Asian J Res Soc Sci Humanit*, **6(8)** (2016) 614–628.

4 Huang G, Liu Z, Van Der Maaten L & Weinberger K Q, Densely connected convolutional networks, *Proc of the IEEE Conf On Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, 4700–4708.

5 Liu R, Jiang Z, Yang S & Fan X, Twin adversarial contrastive learning for underwater image enhancement and beyond, *IEEE Trans Image Process*, **31** (2022) 4922–4936.

6 Guo Y, Li H & Zhuang P, Underwater image enhancement using a multiscale dense generative adversarial network, *IEEE J Ocean Eng*, **45(3)** (2019) 862–870.

7 Yeh C H, Huang C H & Kang LW, Multi-scale deep residual learning-based single image haze removal via image decomposition, *IEEE Trans Image Process*, **29** (2019) 3153–3167.

8 Huo G, Wu Z & Li J, Underwater object classification in side scan sonar images using deep transfer learning and semisynthetic training data, *IEEE Access*, **8** (2020) 47407–47418.

9 Fang F, Li J & Zeng T, Soft-edge assisted network for single image super-resolution, *IEEE Trans Image Process*, **29** (2020) 4656–4668.

10 Deepika P & Pabitha P, Evaluation of convolutional neural network architecture for feasibility analysis on fetal abdomen and brain images, *J Med Imaging Health Inform*, **11(10)** (2021) 2573–2583.

11 Chen K H, Shou T D, Li J K H, & Tsai C M, Vehicles detection on expressway via deep learning: Single shot multibox object detector, in *IEEE Int Conf Machine Learn Cybernet (ICMLC)*, **2** (2018) 467–473.

12 Swarna Priya R M, Gunavathi C & Aarthy S L, Estimating the distance of a human from an object using 3d image reconstruction, in *Informat Syst Design Intel Appl*, Springer, Singapore, 2019, 235–243.

13 HS R K & Bhat D, A novel method to recognize object in images using convolution neural networks, in *IEEE Int Conf Intel Comput Control Syst (ICCS)* 2019, 425–430.

14 ZhaoZ Q, Zheng P, Xu S T, and Wu X, Object detection with deep learning: A review, *IEEE Trans Neural Netw Learn Syst*, **30(11)** (2019) 3212–3232.

15 Emera I & Sandor M, Creation of farmers' awareness on fall armyworms pest detection at early stage in rwanda using deep learning, in *IEEE 8th Int Congress Adv Appl Informat (IIAI-AAI)* (Toyama, Japan) 2019, 538–541.

16 Yao G, Lei T & Zhong J, A review of convolutional-neural-network-based action recognition, *Pattern Recognit Lett*, **118** (2019) 14–22.

17 Huang J, Qin F, Zheng X, Cheng Z, Yuan Z, Zhang W & Huang Q, Improving multi-label classification with missing

labels by learning label-specific features, *Inf Sci*, **492** (2019) 124–146.

18 Jalal A, Salman A, Mian A, Shortis M & hafait F, Fish detection and species classification in underwater environments using deep learning with temporal information, *Ecol Inform*, **57** (2020) 101088.

19 Shen, Z Y, Han SY, Fu LC, Hsiao P Y, Lau Y C & Chang S J, Deep convolution neural network with scene-centric and object-centric information for object detection, *Image Vis Comput*, **85** (2019) 14–25.

20 Malhotra P and Garg E, Object Detection Techniques: A Comparison, in *IEEE 7th Int Conf Smart Structures Syst (ICSSS)* (Chennai, India) 2020, 1–4.

21 Fang W, Wang L & Ren P, Tinier-YOLO: A real-time object detection method for constrained environments, *IEEE Access*, **8** (2019) 1935–1944.

22 Garcia-Dominguez M, Dominguez C, Heras J, Mata E & Pascual V, FrImCla: a framework for image classification using traditional and transfer learning techniques, *IEEE Access*, **8** (2020) 53443–53455.

23 Li R, Wang R, Zhang J, Xie C, Liu L, Wang F, Chen H, Chen T, Hu H, Jia X & Hu M. An effective data augmentation strategy for CNN-based pest localization and recognition in the field. *IEEE Access*, **7** (2019) 160274–160283.

24 Das S, Sharma R, Gourisaria M K, Rautaray S S & Pandey M A, Model for probabilistic prediction of paddy crop disease using convolutional neural network, in *Intelligent and Cloud Computing* edited by D Mishra, R Buyya, P Mohapatra, S P (Series: *Smart Innovation, Systems and Technologies*, Springer, Singapore) 2021, 194.

25 Hastie T, Tibshirani R & Friedman J, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Science & Business Media) 2009, 219–223.

26 Li C, Guo C, Ren W, Cong R, Hou J, Kwong S & Tao D, An underwater image enhancement benchmark dataset and beyond, *IEEE IEEE Trans Image Process*, **29** (2019) 4376–4389.

27 Ronneberger O, Fischer P & Brox T, U-Net: Convolutional networks for biomedical image segmentation, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI, Lecture Notes in Computer Science* edited by N Navab, J Hornegger, W Wells & A Frangi (Springer, Cham) **(9351)** 2015, 234-241.

28 Ahn N, Kang B & Sohn K A, Fast, accurate, and lightweight super-resolution with cascading residual network, *Proc European Conf Comput Vis (ECCV)*, 2018, 252–268.

29 Li W, Liu K, Yan L, Cheng F, Lv Y & Zhang L, FRD-CNN: Object detection based on small-scale convolutional neural networks and feature reuse, *Sci Rep*, **9(1)** (2019) 1–12.

30 Lohia A, Kadam K D, Joshi R R & Bongale A M, Bibliometric analysis of one-stage and two-stage object detection, *Libr Philos Pract*, 4910, (2021) 1–32.

31 Jiang L & Li X, An efficient and accurate object detection algorithm and its application, in *IEEE 5th Informat Technol Mechatron Eng Conf (ITOEC)* (Chongqing, China) 2020, 656–661.

32 Yulin T, Jin S, Bian, G & Zhang Y, Shipwreck target recognition in side-scan sonar images by improved YOLOv3 model based on transfer learning, *IEEE Access*, **8** (2020) 173450–173460.

33 Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M & Berg A C, Imagenet large scale visual recognition challenge, *Int J Comput Vis*, **115(3)** (2015) 211–252.

34 Everingham M, Van Gool L, Williams C K, Winn J & Zisserman A, The pascal visual object classes (voc) challenge, *Int J Comput Vis*, **88(2)** (2010) 303–338.

35 Lin T Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P & Zitnick C L, Microsoft coco: Common objects in context, in European conference on computer vision, (Springer, Cham) 2014, 740–755.

36 Sahu B & Mishra D Performance of feed forward neural network for a Novel Feature Selection Approach, *Int J Comput Sci Inf Tec,* **2(4)** (2011) 1414–1419.