Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Underwater Image Transmission Model Using Improvised Feed Forward Fully Connected Deep Neural Network

Sumanth Sakkara[1], Cyril Prasanna Raj P.[2]

[1]Research Scholar, Department of Electronics and Communication Engineering, VTU-RRC, Visvesvaraya Technological University, Belagavi, India.
[2]Professor, Department of Electronics and Communication Engineering, Cambridge Institute of Technology, Bangalore, India.

E-mail: [1]sumanth.tssg@gmail.com, [1]sumanthsakkara@pes.edu, [2]cyril@cambridge.edu.in

Orcid ID: [1]0009-0000-7095-879X

Abstract

Underwater security surveillance in coastal regions of India is of paramount importance. Underwater drones are deployed and enabled with acoustic sensors and cameras to detect and identify foreign objects. However, reliable underwater image transmission remains challenging due to channel noise and limited underwater acoustic bandwidth. In this paper, a novel method for the transmission of images captured underwater to base stations with minimum loss and robust against underwater noise is developed. The proposed Deep Neural Network (DNN) model, comprising multiple stages of neural network structure, processes the input image and compresses it into a data stream generated using QAM schemes to be transmitted over an underwater channel. The DNN structure at the receiver reconstructs the data stream into a visible image with minimal noise. The developed model was evaluated for performance considering different underwater channel conditions with SNR settings from -10 dB to + 10 dB. The BER metric was estimated to vary between 10^{-2} to 3.4×10^{-3}, with an SSI measurement of 0.9235 and entropy measuring 93.44%. The reconstructed images were analyzed with the PSNR metric, and the proposed model demonstrates an improvement of 51.48%. The proposed framework achieves up to 93.75% compression, reducing bandwidth requirements for underwater acoustic communication while preserving image quality. The simulation results were obtained using the MATLAB environment by modeling the proposed DNN model and training was carried out considering 3590 images. The end-to-end model was evaluated for its performance and effectiveness, demonstrating its suitability for reliable image transmission through underwater channels.

Keywords: Underwater Image Transmission, Underwater Communication, Deep Neural Network, Fully Connected Network, Modulation Schemes.

1. Introduction

The Internet of Underwater Things (IoUTs) is an intelligent network connecting underwater devices, enabling the monitoring of unexplored underwater areas. IoUTs consist of edge devices that acquire data (acoustic, image, video) and transfer this information to

DOI: https://doi.org/10.36548/jiip.2026.3.019
Received: 09.05.2026, received in revised form: 15.06.2026, accepted: 06.07.2026, published: 21.07.2026

destination nodes. The information gathered at these sink nodes is analyzed, monitored, and serviced using on ground support systems [1]. Underwater communication is a challenging technology due to the oceanic environment and its dynamic changes in physical characteristics [2]. The physical layers supporting underwater communication systems are electromagnetic, optical, magnetic induction, and acoustic waves [3]. An acoustic communication system is the most preferred technique for underwater communication [4]. In acoustic communication systems [5], transmission bandwidth is limited (10 Hz to 1 MHz), IoUTs are battery operated edge devices, and theiracoustic communication systems drain power quickly, with no provision for battery recharging in underwater environments [6]. For IoUTs to be commercially available, edge devices that consume low power and support high data rate transmission for all real-time underwater operations are needed [7]. Transmission of high-bandwidth consuming data, such as images [8] and video signals, need to be compressed first, and the compressed data must then be modulated to meet underwater acoustic channel requirements [9, 10]. Exploration of underwater environments is carried out using Unmanned Underwater Vehicles (UUVs) that capture images underwater as they navigate [11]. UUVs are retrieved, and the acquired images are downloaded by the operator for further analysis. However, the actual monitoring of the underwater environment by the operator is delayed, and revisiting the same area is challenging [12]. Modern naval mine counter missions [13] utilise UUVs and Unmanned Surface Vehicles (USV) with in situ processors as payloads for data processing and target detection. A snippet of the target detected by the payload is transmitted by the UUV to USV along with geo-referencing of its location [14]. The operator decides to revisit the target (operates the UUV) with the geo-referenced information for image/video recording of the target, which is observed once the UUV is retrieved. The machine learning model [15] for symbol detection with channel estimation uses a neural network approach and has demonstrated that the neural network model is simpler than CNN and RNN models for data transmission in an underwater environment. In the OFDM model, at the receiver side, a channel compensation model was introduced after the FFT model to improve the Symbol Error Rate and Bit Error Rate.

Doppler shifts, scattering, multi-path, variations in sound speed, and ambient noises from natural and anthropogenic sources are the main contributors to the degradation of underwater acoustic communication quality [16,17]. These parameters also impact the unpredictability of signal behavior. Further increasing complexity is the demodulation of information from the modulated signals [18]. In addition, variations in the velocity of sound, multi-path signals, acoustic noise from aquatic animals and humans, and roughness in the seabed complicate the estimation of channel characteristics [19,20]. Most underwater communication systems [21] use channel estimation algorithms to estimate channel state information by using a pilot signal that is transmitted and estimated at the receiver, thus providing information on channel noise [22, 23]. Computing channel characteristics is a computationally intensive process and an additional overhead, and hence introduces delay [24]. Accurate estimation of channel characteristics is an important criterion for better Bit Error Rate (BER) performances of the communication system [25]. AI and deep learning methods have been extensively adopted to overcome the challenges of conventional methods. AI models are trained to detect the message directly from the data. These models are robust even with the dynamic nature of channel characteristics conditions [26 to 28]. A detailed study on AI methods for communication systems is presented in [29]. Most of the methods reported demonstrate improvement in BER performances as compared with conventional methods [30]. Deep Neural Network (DNN) models accurately estimate the channel characteristics in real underwater environments and have demonstrated that Long Short-Term Memory (LSTM) is the most appropriate model with mean absolute percentage error for underwater channels. In [31,32], the authors developed a DNN model trained to recover the transmitted signals at the receiver without any dependency on channel characteristics due to prior training of the model. The trained DNN model was introduced at the receiver stage of the OFDM based communication model. The BER results were in the order of 10-3 at 20 to 25 dB Signal to Noise (SNR) range and offered higher spectral efficiency.

This paper introduces a lightweight deep learning model for an underwater image transmission (UWIT) framework. The DNN encoder-decoder model was used for image compression and reconstruction over impaired underwater acoustic channels. The image transmission model was evaluated across multiple modulation schemes and realistic noise conditions. A dedicated DNN enhancement network denoises reconstructed images, improves visibility for object detection, and preserves entropy close to ground truth images. The framework achieves reliable image reconstruction even under low SNR underwater conditions.

The remainder of this paper is organized as follows. Section 2 describes the overall DNN framework for UWIT, the underwater acoustic channel model and the encoder-decoder DNN architecture. Section 3 presents the proposed DNN-based encoding-decoding image transmission model. It covers DNN architecture training process, configuration, dataset preparation and the development of an improvised DNN enhancement model for UWIT. Section 4 discusses the experimental results, highlighting the performance of different DNN configurations and compression algorithms. Section 5 concludes the paper with a summary of research findings and directions for future work.

2. Deep Neural Network Method and Underwater Acoustic Channel

The overall image transmission framework for the Underwater Acoustic (UWA) communication system, using a DNN-based encoding and decoding model, is illustrated in Fig. 1. The input images captured underwater are preprocessed and resized to N1×N2, where N1, N2∈ {32, 256, 512, 1024}. The resized image is then divided into non-overlapping subimages of size b1×b1, where b1∈{4, 8, 16, 32}. Each sub-image of size b1×b1 is then reordered into a one-dimensional (1D) vector of length b12×1. The trained DNN encoder processes this 1D vector to generate compressed output of size m1×1 (m1<<b12 and m1∈ {4, 8, 16}). The encoded outputs from all sub-images are converted into binary data using Huffman coding and are mathematically modulated in MATLAB using five standard modulation schemes: Frequency Shift Keying (FSK), Phase Shift Keying (PSK), QPSK (Quadrature Phase Shift Keying), QAM-16, (Quadrature Amplitude Modulation-16 and QAM-64 Quadrature Amplitude Modulation-64 (QAM-64)]. The BER performance of each scheme is evaluated under varying signal-to-noise ratio (SNR) conditions. The underwater channel propagation effects are incorporated through the analytical UWA channel model. At the receiver the demodulated binary data yield the estimated vector $\widehat{\mathrm{m}}1$, which is then processed by the trained DNN decoder to reconstruct $\hat{\mathrm{b}}12$. The primary focus of this work is the lightweight DNN-based encoder and decoder for image encoding and reconstruction. Finally, the reconstructed subimages are regrouped to obtain the restored image at the receiver.

In underwater image transmission, signal attenuation occurs due to absorption and scattering by water molecules and suspended particles, resulting in rapid intensity loss, especially for high-frequency acoustic or short-wavelength signals. This leads to faded or low-contrast images. Noise originating from marine life, ship engines, and sensor electronics introduces random variations that corrupt pixel and bit data, thereby reducing image reconstruction quality. Multipath effects caused by signal reflections from the seabed, surface and underwater objects produce delayed multipath signal componenets, leading to image distortion and synchronization errors. These factors degrade image reconstruction quality, thereby requiring adaptive modulation, equalization, and error-correcting codes to ensure reliable underwater image transmission. DNN training enhances underwater image transmission by learning nonlinear mappings that mitigate attenuation, noise, and multipath effects. DNN models can perform channel estimation, denoising, and image restoration,thus improving reconstruction quality and enabling adaptive equalization without the need for explicit physical channel modeling. In this work, a conventional feedforward neural network is extended into a deep neural network to effectively capture the complex nonlinear relationships between the transmitted and received signals. The DNN learns to predict attenuation characteristics, suppress noise components, and mitigate multipath distortions, resulting in accurate image reconstruction and efficient channel equalization in underwater acoustic environments.

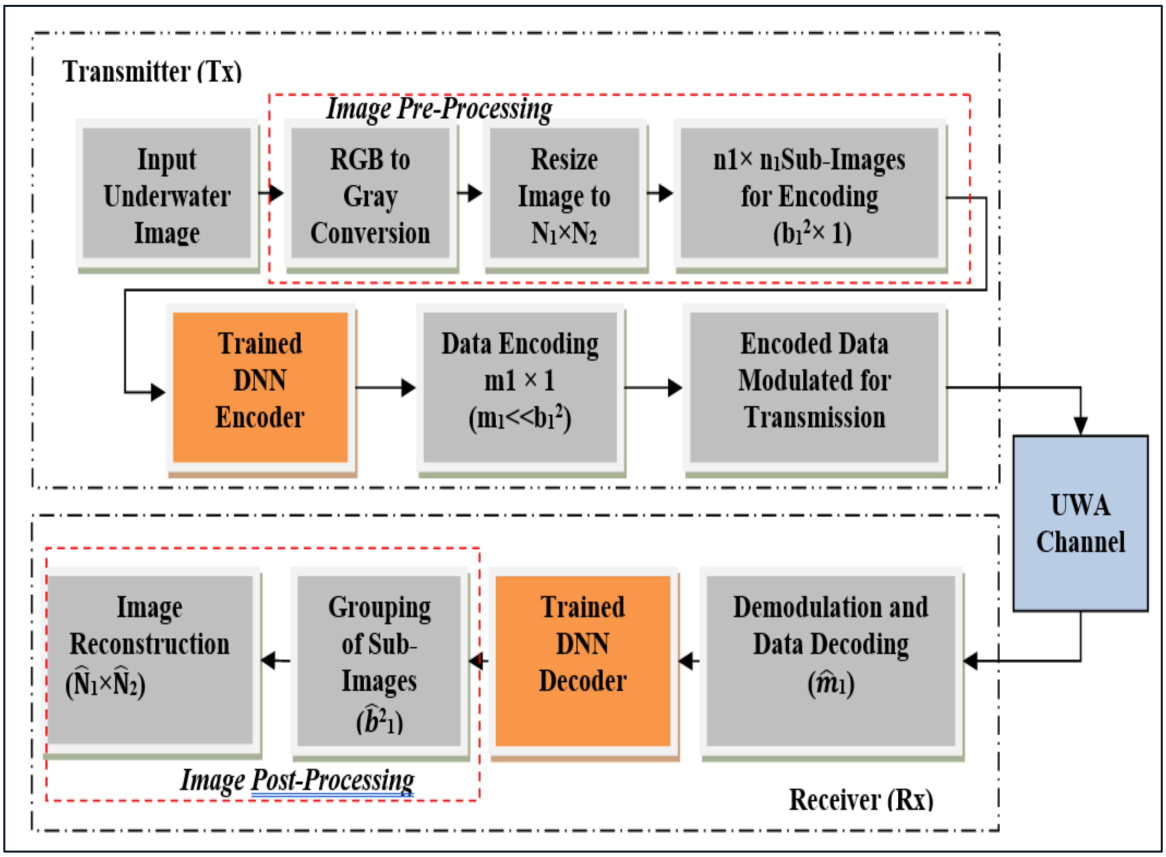


Figure 1. Image encoding and decoding transmission framework for underwater acoustic communication

2.1 Deep Neural Network Method

DNN is a model with an input layer, multiple hidden layers, and an output layer. Each layer consists of multiple neurons trained to learn the projection of information from the previous layer to the next layer. This projection is a non-linear function represented by the weighted sum of the neuron activity of the preceding layer. The structure of the DNN is presented in Fig.2, the model has L layers, and the output of each layer is expressed as:

$$S_i = \sigma\,(W_i\, S_{i-1} + b_i), \qquad I = 1,2,3,\ldots,L \tag{1}$$

Where the parameters σ in Eq. (1) are a non-linear function such as sigmoid, tanh, softmax, ReLU etc., and θi= {Wi, bi} and {θi, θi+1, ….., θL } are the parameters of the neurons and layers. The weight and bias vectors are represented as Wi× Si−1, bi. Considering the input as image with size N1×N2, where N1, N2 ∈ {32, 256, 512, 1024}, and the input layer (S0) of DNN model comprising of M inputs, the N1×N2 image is grouped into b1× b1 blocks, where b1 ∈ {4, 8, 16, 32}. The sub image of b1×b1 is rearranged into 1D matrix of size M which forms the input to the DNN at the input layer. M-vectors of the input layer are processed by multiple hidden layers (S1, S2, ….., SL-1) and are finally available as input to the output layer (SL) that generates the final output. The input layer vectors are defined as P1, P2, P3, ……PM, connected to the hidden layer S1 through the Network Connection Layer (NLC). The NLC is a network architecture that connects every data from the preceding layer to the input of a neuron in the next layer. In the network model shown in Fig.2, there are three hidden layers (S1, S2 and S3), one input layer S0 and one output layer SL. The number of neurons in the S1 layer are n11, n12, n13,… n1a1 similarly for S2 and S3 hidden layers. The output layer SL has M outputs denoted as O1, O2, O3,…,OM.

The neurons of each layer are represented as nkak, where k is 1, 2, 3,.., L. The number of hidden layers and the number of neurons in each hidden layer are identified based on the requirements and applications. In this work, the input layer and output layer will have a data vector of size M. The network model is defined as M:a1:a2:a3:M, indicating that there are M inputs in the input layer, a1, a2 and a3 neurons in hidden layers 1, 2 and 3 respectively and M neurons or outputs in the output layer.

The network model combinations are 16:4:16, 64:8:64, 256:16:256, 64:32:8:32:64. The hidden layer of the lowest size ensures that the input data of size M is mapped to K symbols (K is 4 or 8) and the encoded K symbols are reconstructed to M symbols at the output layer. Supervised training is carried out by setting the input vector as the desired output Y = [Y1, Y2, Y3, Y4, …, YM] at the output layer. The network elements such as weights and biases are identified during the training process such that the error between the desired output and the network output O = [O1, O2, O3, …, OM] is minimal or nearly zero. The training data set is represented as in Eq. (2):

$$\{(P1, Y1), (P2, Y2), (P3, Y3),\ldots(PM,YM)\} \tag{2}$$

The number of training dataset is considered to be H with M vectors in each data set. It is desired to achieve the minimum loss function between the desired output and the actual output for all training datasets using stochastic gradient decent and backpropagation during training.

Fig.3 presents the proposed network model for underwater image transmission (UWIT) with a configuration of 64: 32: 4: 32:64. The input vector of 64 elements is encoded to 4 outputs in the second hidden layer (S2) and decoded to 64 elements in the output layer (SL). There are two intermediate hidden layers, S1 and S3, that are sandwiched between the S0 & S2 layers and S3 & SL respectively.

Table 1 presents the network parameters for the 64:32:4:32:64 model, which provides information on the number of weights and biases at each layer, along with the Network Activation Function (NAF). With training, the network weights and biases are identified in phase 1 of the activity. In phase 2 of activity, the trained network is split into two models: the transmitter, and receiver as shown in Fig. 4. The transmitter (encoder, Fig. 4(a)) comprises the S0, S1 and S2 layers, and the receiver model (decoder Fig. 4(b)) comprises the S3 and SL layers. The weights and biases are loaded into the encoder and decoder networks to process the input frames. The encoded data, or the output of the S2 layer, consists of 4 symbols grouped into packets of 128 symbols with Start of Frame (SOF) and End of Frame (EOF), symbols of all zeros (0x00) and ones (0xFF) respectively.

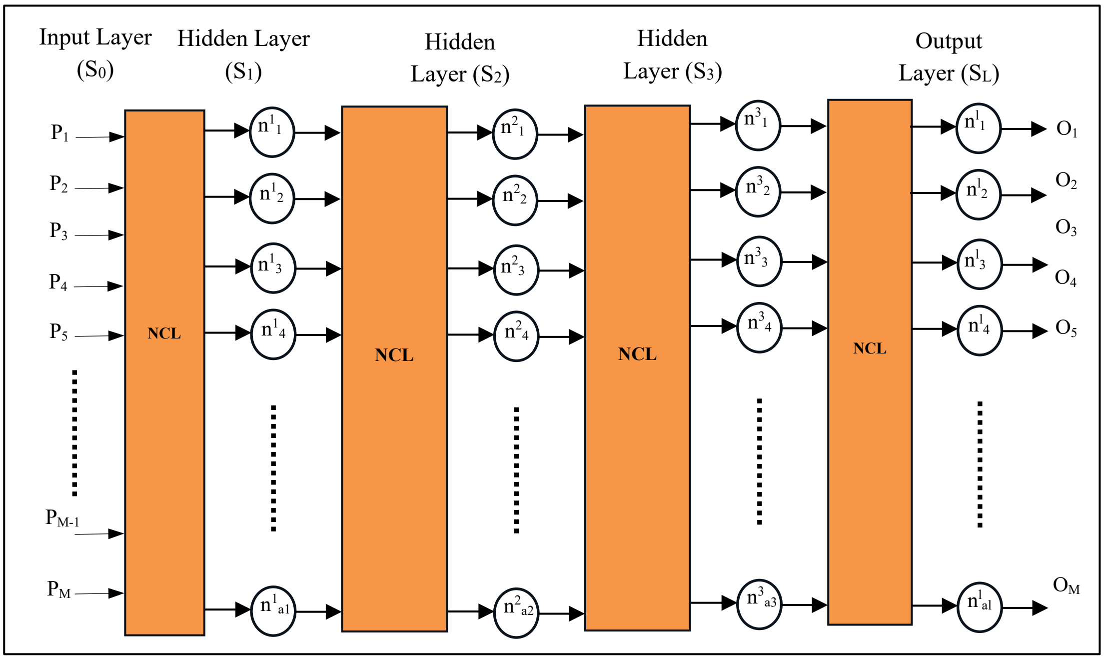


Figure 2. The structure of DNN model

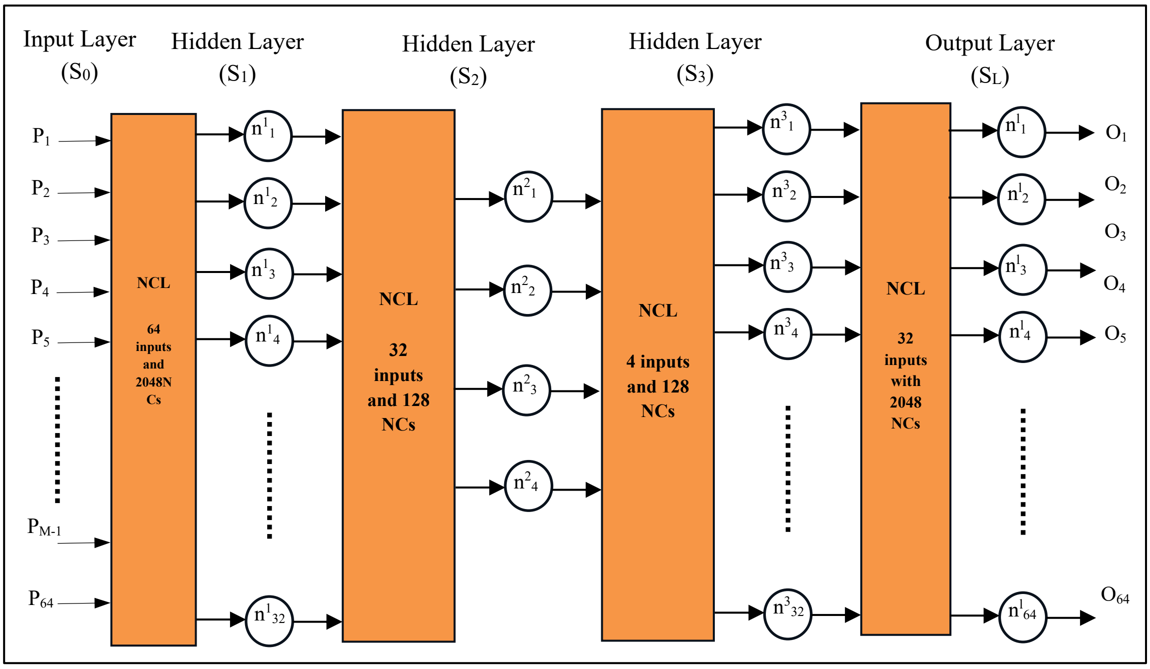


Figure 3. DNN model with 5 layers for encoding and decoding for underwater image transmission

The selection of network sizes is based on the compression ratio as defined inEq. (3). Where I_1 and I_2 represent the input and bottleneck feature dimensions, respectively:

$$r = \frac{I_2}{I_1} \tag{3}$$

This ratio is chosen to preserve the mutual information between the input P and its latent representation O, as shown in Eq.(4). The bottleneck dimensionI_2must be sufficiently large to retain critical structural and semantic information, while remaining compact enough to remove redundancy:

$$I(P;O) \approx H(O) - H(O \mid P) \tag{4}$$

Table 1. Network model configuration

Network	64: 32: 4: 32: 64 configurations No. of input and output vectors = 64		
	No. of weights	**No. of biases**	**NAF**
S_0 (input layer)	-	-	-
S_1 (Hidden layer 1)	2048	32	ReLU
S_2 (Hidden layer 2)	128	4	Softmax
S_3 (Hidden layer 3)	128	32	Sofmax
S_L (output layer)	2048	64	ReLU

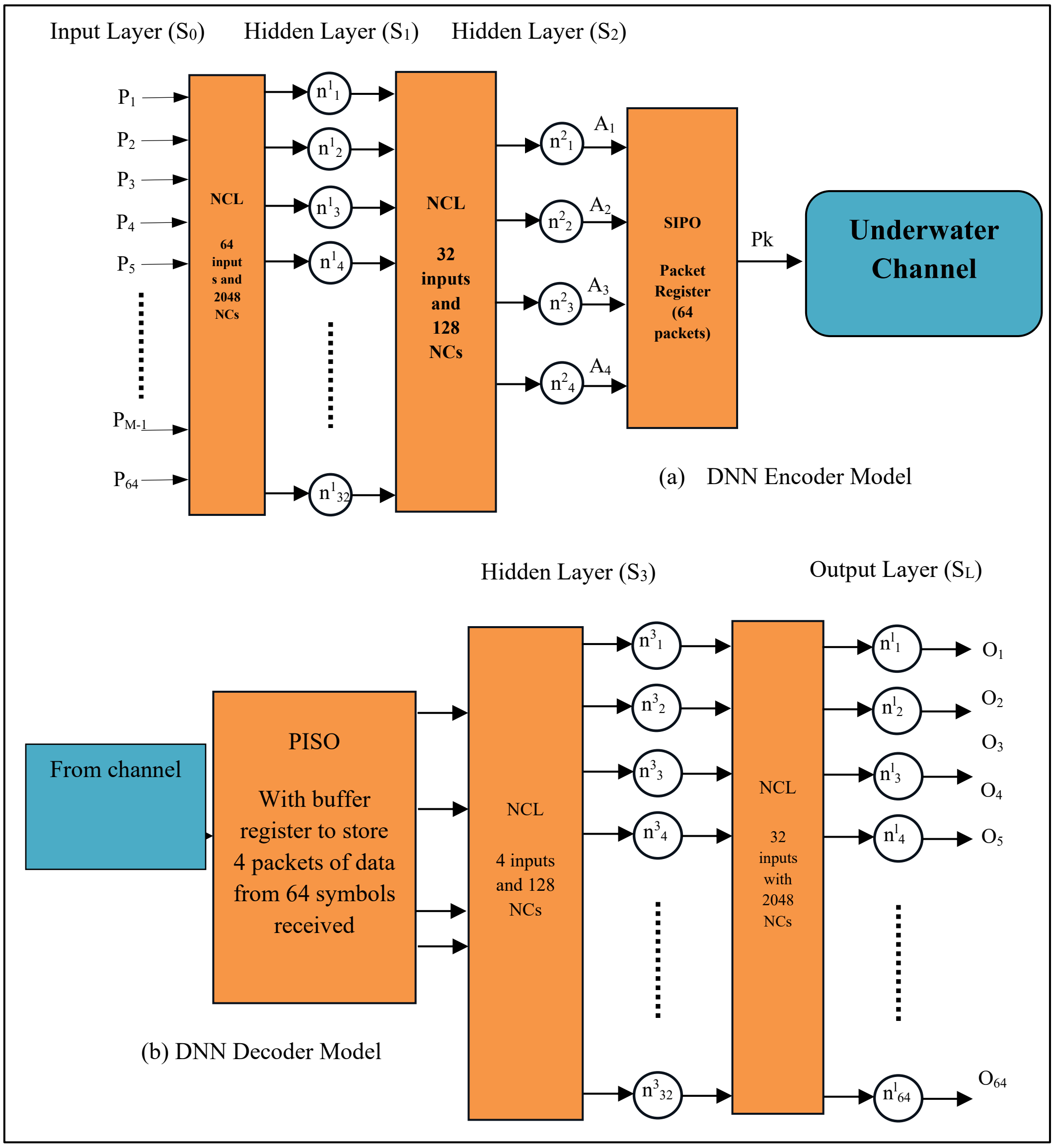


Figure 4. a) DNN encoder model and b) DNN decoder model for underwater communication

$$L = \| P - \hat{P} \|^2 \tag{5}$$

In practice, I_2 is tuned so that the reconstruction loss L in Eq. (5) remains below a target threshold ϵas specified in Eq. (6):

$$\epsilon \in [5 \times 10^{-3}, 1 \times 10^{-2}] \tag{6}$$

A smaller ε corresponds to improved reconstruction accuracy and reduced information loss. For underwater imaging, a practical and robust choice is which allows effective compression while maintaining perceptually acceptable image quality under typical underwater noise and scattering distortions. Equations (3) to (6) collectively guide the selection of the network bottleneck size to balance compression efficiency and reconstruction accuracy.

2.2 Modelling of Acoustic Channel

A channel model for an underwater environment is estimated to compute the Packet Error Rate (PER) which is based on BER and SNR. As acoustic waves propagate best underwater, there are constraints such as propagation delay, SNR, spreading loss, propagation sound, ambient noise and absorption loss. SNR in underwater is computed considering four factors: source level (Slevel), transmission loss (Tloss), noise level (Nlevel) and directivity index (Dindex).

Considering these four components SNR is given as in Eq. (7):

$$\gamma = S_{level} + T_{loss} + N_{level} + D_{index} \tag{7}$$

Where, γ represents the SNR of an underwater channel. Source level noise, transmission loss, and noise level are expressed in Eq. (8), Eq. (9) and Eq. (10) respectively. Noise level is expressed in terms of turbulence noise, shipping noise, wave noise, and thermal noise. The DI noise is set to zero, assuming an underwater hydrophone with an omni-directional pattern.

$$\mathrm{S_{level}} = 10\ [\log(\mathrm{p}) - \log(4\pi \mathrm{r}^2) - \log(0.67 \text{ x } 10^{-18}\ )] \tag{8}$$

$$\mathrm{T_{loss}} = 20 \log(\mathrm{d}) + \alpha(\mathrm{f})\ \mathrm{d} \text{ x } 10^{-3}\ ) \tag{9}$$

$$\mathrm{N_{level}} = 50 - 18 \log(\mathrm{f}) \tag{10}$$

Combining all the four parameters, SNR of underwater channel is expressed as in Eq. (11):

$$\gamma = 10\ [\ \log(\mathrm{p}) - \log(2\pi \mathrm{r}^2) - \log(0.67 \text{ x } 10^{-18}) - 20 \log(\mathrm{d}) - \alpha\ (\mathrm{f}) \text{ x d x } 10^{-3} - 50 + 18 \log(\mathrm{f}) \tag{11}$$

The parameters P, r, d, f, and α(f) represent transmitter power in Watts, the radius of the sphere or region of interest, the distance between the nodes, the operating frequency of the transmitter-receiver module, and the absorption coefficient (dB/km), respectively. The average BER is expressed as in Eq. (12) for shallow water and Eq. (13) for deep water.

$$\mathrm{BER}\ (\gamma) = \frac{1}{2}(1 - \sqrt{\frac{10^{\left(\frac{\gamma}{10}\right)}}{1+10^{\left(\frac{\gamma}{10}\right)}}} \tag{12}$$

$$\text{BER}(\gamma) = \frac{1}{2}\ \text{erfc}\sqrt{\frac{\text{K}10^{\left(\frac{\gamma}{10}\right)}}{\text{K}+10^{\left(\frac{\gamma}{10}\right)}}} \tag{13}$$

The BER is a measure of the number of bit errors perunit of time in a given communication channel. From the expressions of BER, PER is computed as in Eq. (14), where Y represents the number of bits considered in a given packet of data:

$$PER = 1 - (1 - BER)^{y} \tag{14}$$

The NOF1 watermark dataset was used for simulation, collected in a fjord environment in June. The transmitter and receiver were bottom deployed at a 750 m range with a water depth of 10 m. The probe used had a -3 dB SNR frequency band of 10 to 18 kHz. Delay coverage was 128 ms, Doppler coverage 7.8 Hz, and a SISO configuration with a single hydrophone was employed. The sounding duration was 32.9 s, cycle time 400 s, and total play time 33 min.

For MATLAB simulations, the underwater channel was configured with 2–4 kHz bandwidth, carrier frequencies of 15–40 kHz, 256 subcarriers, and OFDM symbol sizes from 62.5–128 ms. The compressed images were transmitted using FSK, PSK, QPSK, QAM-16, and QAM-64 modulation schemes with Huffman encoding. A neural network-based FCNN model was used for block-level encoding and decoding, enabling efficient image compression and reconstruction over the underwater acoustic channel.

3. Proposed Work

The communication module for underwater image transmission is a complex system that operates in dynamic environmental conditions. These include non-stationary water conditions, the presence of bubble layers, varying levels of seawater turbidity, ocean noise, different seafloor types, and inhomogeneous objects. The underwater channel is time-variant, has limited bandwidth with multipath effects, and causes long delays in signal transmission.

3.1 DNN Based Encoding-Decoding System for UW Image Transmission

The trained DNN model was evaluated by providing the test dataset to the NN encoder, which compresses the input data into a lower-dimensional latent space. The compression ratio varied depending on the selected network architecture and size, reflecting the capacity of the trained model. The resulting compressed representations are Huffman encoded, and the encoded data bits are transmitted over the UW channel with different noise conditions using suitable modulation schemes such as FSK, PSK, QPSK, 16-QAM, and 64-QAM. The underwater channel was mathematically modeled in MATLAB as described in Section 2.2, incorporating various attenuation and distortion effects typical of underwater communication. At the receiver, the compressed data was decoded and decompressed using the trained DNN decoder to recover the data. The decompressed vectors were reshaped into $b_1 \times b_1$ blocks, which were then used to reconstruct the full $N_1 \times N_2$ grayscale images. The reconstructed images were compared against the ground truth images using PSNR and SSI metrics to evaluate the fidelity and robustness of the image reconstruction process under channel impairments.

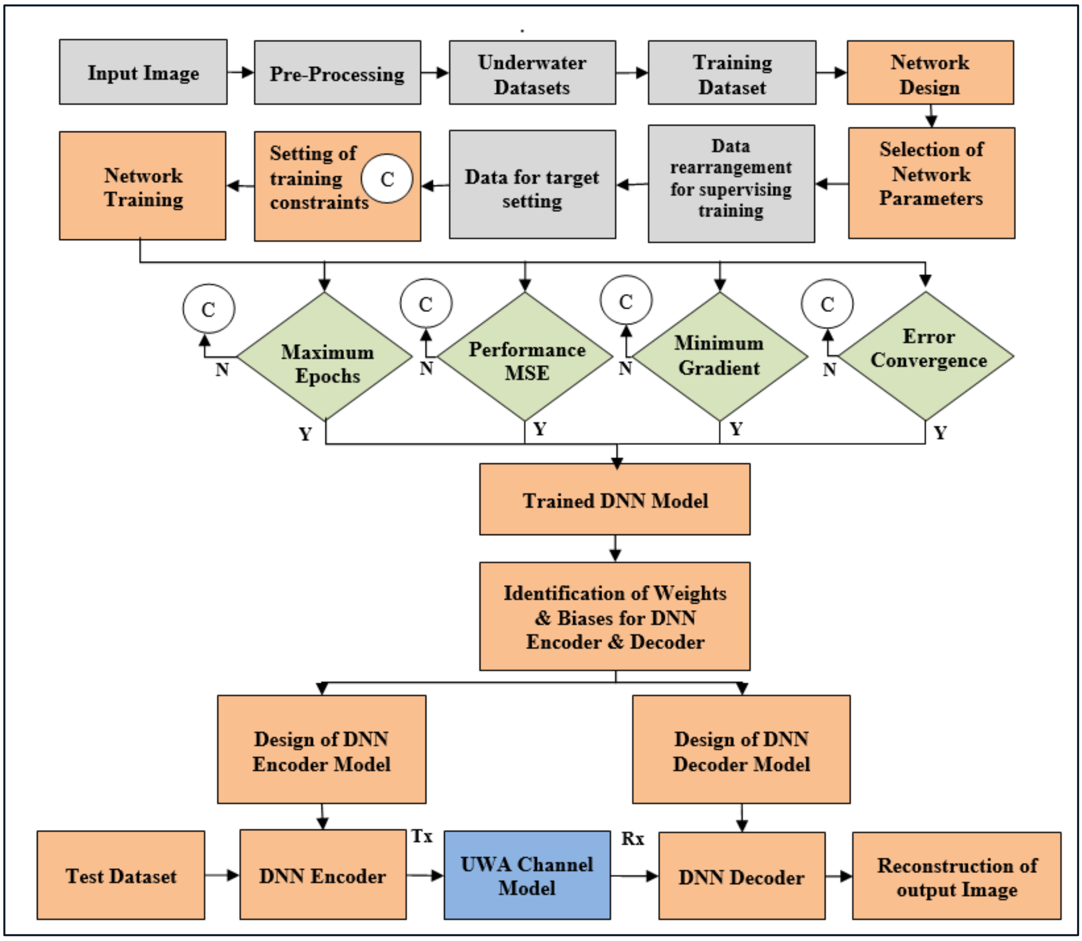


Figure 5. System-level training and evaluation framework of the proposed underwater DNN encoder–decoder model

3.2 Underwater Image Dataset

The neural network was trained by setting the input and target through supervised learning, where the target and input matrices are the same. The training process is completed once the constraints and parameters are met. NN encoder and NN decoder were designed by extracting weights and biases from the trained neural network model. The NN model was tested by providing the test dataset to the NN encoder to compress the data into a lower latent space. The compressed image was encoded using the Huffman model, and the encoded data bits were transmitted over the channel using suitable modulation schemes. At the receiver, the compressed data was subsequently decompressed by the trained NN decoder back to its original size format. The decompressed data was then converted from vectors to blocks, reconstructing the initial non-pixel blocks. These blocks were used to reconstruct the 32x32 grayscale image. Training of the NN model was carried out on a dataset of underwater images. The training dataset comprised synthetic underwater images incorporating multiple marine objects such as fish, plants, and tortoises. These synthetic images have variations in texture, structure and color enabling the network to learn generalized features across multiple object categories. The images were generated under varied underwater conditions, including changes in turbidity, illumination, and depth to emulate realistic optical attenuation and scattering effects. Prior to training the images underwent preprocessing operations such as normalization, noise modeling and contrast enhancement to align their statistical characteristics with those of real underwater acquisitions and improve the robustness of the DNN model.

Each fish species included in the dataset had a variety of color combinations, textures, and geometrical dimensions. Three fish images were oriented to the right, while the other ten images were headed to the left. Fish_1 and fish_6 have separate fish head, eye, and mouth angles and orientations. Each fish had distinct pelvic fin, anal fin, and caudal fin lengths, and colors. The majority of the fish species in the dataset featured grey fins, white eyes, and black bodies. The length of the body, pelvic fin, anal fin, caudal fin, and other fish characteristics are extracted geometrically. The color pattern (grey, white and black texture) and other aspects are also included in the feature extraction based on appearance. Such features, however, are susceptible to variations in illumination levels and noise. Ten different underwater tortoise images were considered.

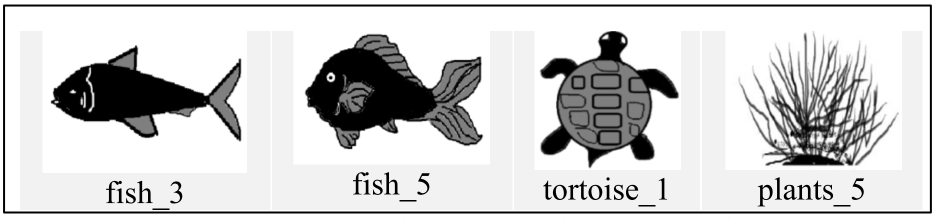


Figure 6. Image dataset for training DNN model

During underwater imaging, cameras mounted on autonomous vehicles or operated by divers are frequently subjected to orientation changes caused by water currents,vechile motion, and irregular terrain. To address orientation changes commonly encountered in underwater image acquisition, rotation-based data augmentation was incorporated during training using MATLAB's imageDataAugmenter function. The augmentation configuration included random rotations within ±30°, horizontal and vertical translations of up to ±20% of the image dimensions, shear deformation, zoom scaling between 0.8 and 1.2, and horizontal reflections.

The applied augmentation parameters approximate variations in pitch, roll, and yaw encountered during underwater imaging, enabling the DNN models to develop rotation-invariant feature encodings. By exposing the model to such orientation variability during training, the system becomes more resilient to arbitrary camera poses, resulting in improved recognition performance and image reconstruction accuracy under dynamic real-world underwater environments.

The UW image dataset is expanded by employing an image augmentation approach. This approach involves modifying the original fish images by rotating them at different angles, shifting them, flipping them, and cropping them. Following the augmentation, each image was converted into 120 images, yielding 1200 images of fish. This approach increases dataset diversity, provides extensive training data, and reduces overfitting. A similar image augmentation approach was used on 10 different tortoise and 10 different UW plant images to create a dataset. Finally, the UW dataset has around 3590 images for training the model. Each of these images considered for training was grouped into 4 categories to evaluate the performances of the proposed models. Entropy values were determined for different images in a dataset that comprises distinct categories such as fish, tortoise, plants, corals, and rocks. The entropy values, ranging from 2.256 to 5.4413, were graphically displayed using a bar chart in Figure 7 emphasizing variations among these categories. This analysis used entropy to measure the visual diversity and amount of information present in each image. The dataset is divided into four groups based on the entropy values as shown in Table 3.

Table 2. Classification of dataset into four groups based on the entropy values

Groups	Entropy Range	Image Dataset
Group-I	2.256 – 2.966	fish_8, fish_3, tortoise_4, fish_10, fish_6, fish_9
Group-II	3.079 – 3.8127	fish_2, fish_5, fish_7, fish_4, plants_3, fish_1, corals_7, tortoise_2, rocks_8, corals_9
Group-III	3.9472 – 4.0636	tortoise_3, tortoise_9, tortoise_5, corals_10, tortoise_10,tortoise_1, tortoise_6, plants_4, plants_2
Group-IV	4.7183 -5.4413	tortoise_7, plants_1, plants_6,tortoise_8, plants_5

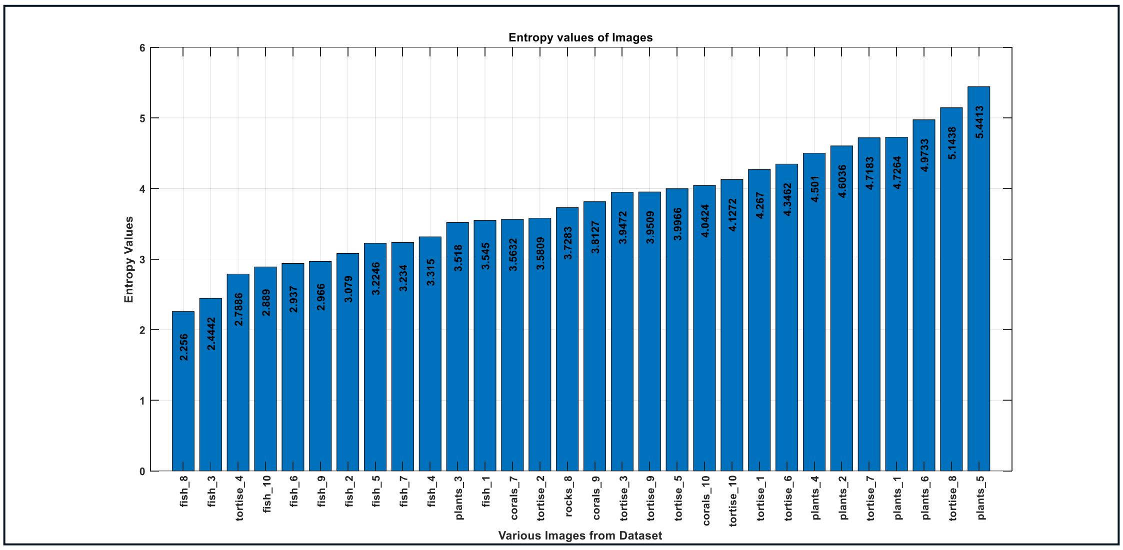


Figure 7. Entropy values of various images from the test dataset graphically represented in bar chart

The four entropy-based underwater image groups exhibit distinct sensitivities to preprocessing operations. Histogram equalization significantly increases entropy, particularly in low-entropy Group I images. As a result, these images may shift to higher groups, whereas high-entropy Group IV images remain largely unaffected. Contrast Limited Adaptive Histogram Equalization (CLAHE) introduces smaller and more localized adjustments, thereby preserving overall group consistency. Color-to-gray conversion generally reduces the entropy of underwater images because chromatic information is removed. The degree of reduction depends on color complexity and the chosen conversion method. Average-based conversion methods reduce more than luminance-weighted methods. Images falling near the transitions between Groups 1–2, 2–3, and 3–4 show the highest sensitivity. Using standardized preprocessing and luminance-preserving conversions ensures more consistent entropy grouping across underwater image datasets.

3.3 Improvised DNN Model for UWIT

The reconstructed images will exhibit checkerboxes since they are grouped into sub-images and, when compared with the input image, will show channel noise. To improve visual clarity and reconstruct the image with fewer distortions a new scheme is proposed. Figure 8 presents the block diagram of the proposed model.

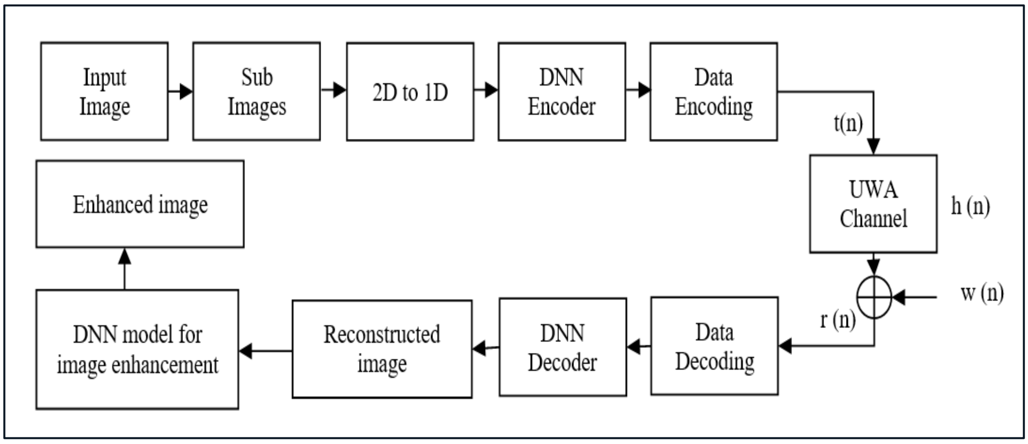


Figure 8. Improvised DNN model for underwater image transmission

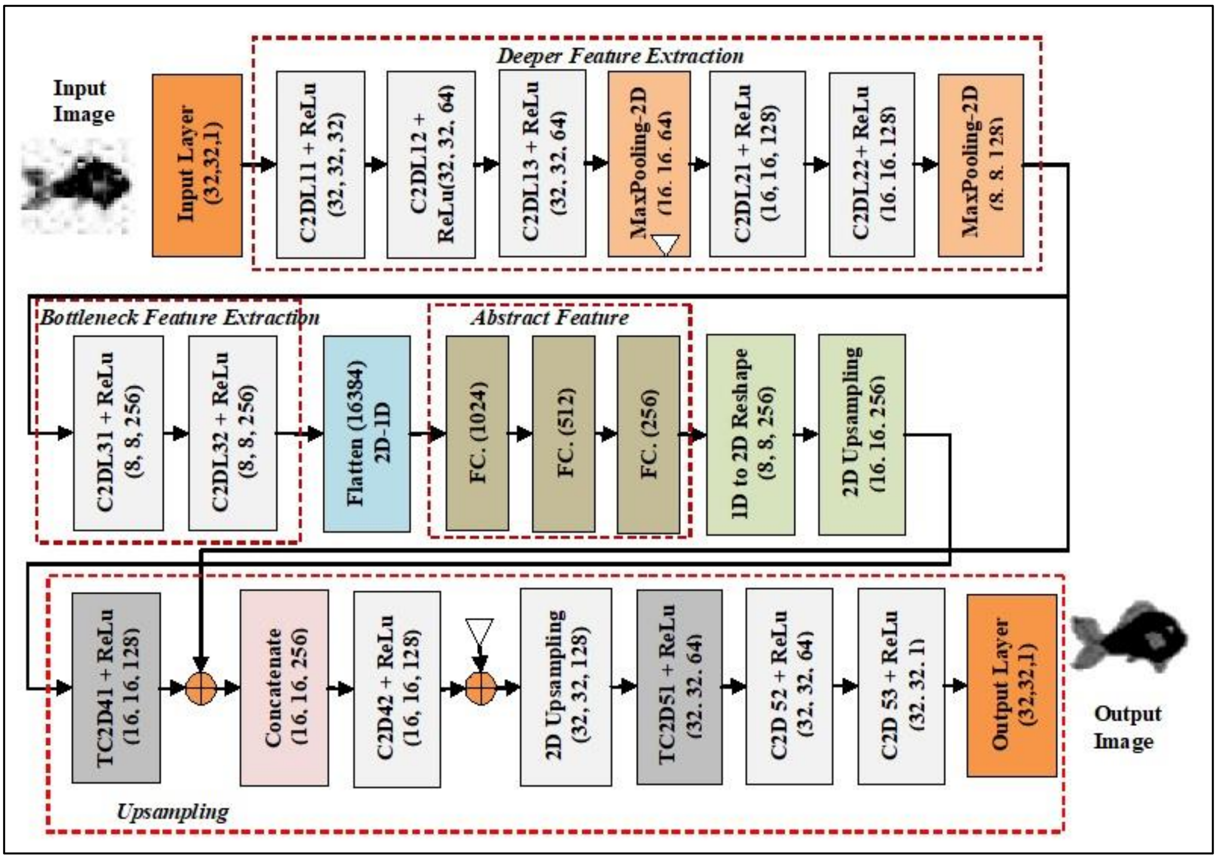


Figure 9. CNN-FC based image enhancement model for noisy underwater images

Deep neural network architecture is used to enhance images for better performance. Convolution Neural Network (CNN) image enhancement model, shown in Figure 9, is trained with noisy reconstructed images obtained from a 16:4:16 DNN decoder model as input dataset and expected clear images as the output dataset. The reconstructed 32x32 grayscale noisy images were processed by multiple two-dimensional convolution layers (C2DL11, C2DL12, C2DL21 and so on) with RelU activation function to extract deep and low-level features, as shown in Fig 8. Max pooling layers were used to reduce the spatial dimensions. The bottleneck captured abstract representations with 256 filters. Fully connected (FC) layers refined these features globally before reshaping them back to 2D for upsampling. Skip connections merged encoder features with decoder layers to preserve spatial details, while upsampling and transposed convolution layers (TC2D) restored the original image size. The input dataset consists of reconstructed images from the decoder with various SNR conditions and is augmented to increase the number of images in the dataset. The model was trained using MSE loss over 50 epochs with the data split into 80% for training and 20% for validation.

4. Results and Discussion

Table 3 summarizes the training results of four lightweight, fully connected, feed-forward DNN networks for the proposed underwater image transmission (UWIT) model. All networks showed good regression, indicating effective feature retention during nonlinear mapping compression.

Table 3. Training results of feed-forward fully connected DNN models for UWIT

Neural Network	Performance	Epochs	Gradient	Regression
16:4:16	0.001594	130	5.3424e-08	0.94821
64:8:64	0.0013081	193	9.9742 e-08	0.95777
256:16:256	0.00079921	353	2.6185e-05	0.97466
64:32:8:32:64	0.0013218	508	1.0185 e-07	0.95734

MATLAB regression plots in Figure 10 show the fit between predicted and target outputs for all networks. The 256:16:256 network exhibits the lowest training error and strongest regression, while the smaller 16:4:16 and 64:8:64 networks converge steadily with moderate accuracy. The 64:32:8:32:64 networks perform similarly to the mid-sized configuration. All networks process images block-wise (sub-images) as discussed in Section 2, allowing them to handle any input image size suitable for real-time underwater deployment.

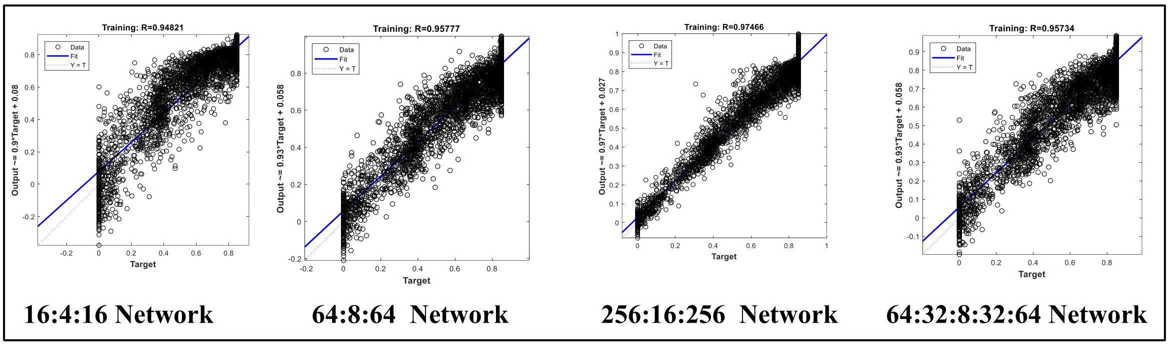


Figure 10. MATLAB regression plots for different DNN networks

In this work, a DNN model was analysed. The input image was resized to 32 x 32 and its histogram along with the entropy was analysed, as shown in the Figure 11. The resized images visually represent underwater objects with varying histograms and entropy levels. The resized input image was reordered into 1D data by considering 4 x 4 sub images and for each 4 x 4 sub-image, a column vector of 16 elements was generated. Arranging the 32 x 32 image into a column vector generated 16 x 64 matrices which were processed by the NN model to generate output data of size 4 x 64 for each input image. The 4 x 64 data was encoded using Huffman coding to generate 64 data elements. The encoded data was modulated using FSK, PSK, QPSK, QAM-16 and QAM-64 modulation schemes and is prepared for transmission over an underwater channel.

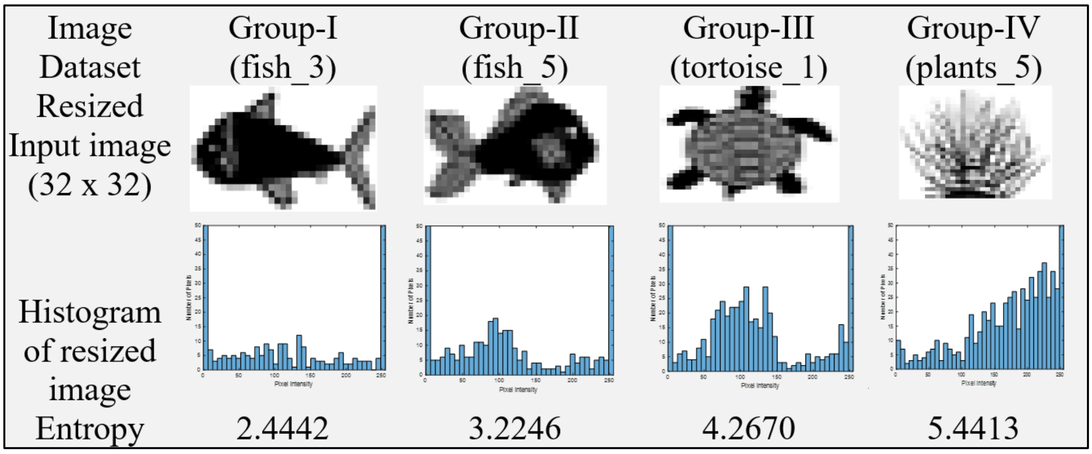


Figure 11. Entropy values of various images from the test dataset graphically represented in bar chart

The encoded data was convolved with an underwater channel and decoded at the receiver end. From the received bits, an inverse operation was carried out to generate 4 x 64 elements, which were then processed by the NN decoder to generate 16 x 64 matrices. A 1D to 2D conversion was carried out, and the image was reconstructed. Performance metrics were computed considering the input image and the reconstructed image. Performance metrics were computed for different modulation schemes. BER values for four different image datasets under varying SNR conditions (-10dB, 0dB, 10dB) using FSK, PSK, QPSK, QAM-16 and QAM-64 modulation schemes are shown in Figure 12.

The BER performance of neural network 1 (16:4:16) for four images of the dataset is shown in Fig. 10, For the fish_3 image with 2.4442 entropy value, at -10 dB SNR, the BER improved by 2.53×10^{-1} when the modulation scheme changed from FSK to QAM-64. Similarly, at a +10 dB SNR, changing the modulation technique from QAM-64 to FSK 1.94×10^{-1} resulted in a 5-fold improvement in BER, contrary to the earlier scenario. When analyzing the BER values for QAM-64 over the SNR range of -10 dB to +10 dB, the results suggested a minimal effect of SNR, with a sensitivity of 3.4×10^{-3} for QAM-64 and 25.7×10^{-3} for FSK modulation. Similarly, for the plants_5 image having a larger entropy of 5.4413 BER value improved by 1.59×10^{-1} for QAM-64 and 1.62×10^{-1} for FSK. At 0dB SNR, all five modulation schemes converge at an average BER value of 3×10^{-1}.

Figure 13 shows the BER performance for the neural network 2 (64:8:64) architecture for four images from the dataset. At -10 dB SNR, the BER improved by 2.37×10^{-1} when the modulation scheme changed from FSK to QAM-64. Similarly, at a +10 dB SNR, changing the modulation technique from QAM-64 to FSK 1.737×10^{-1} resulted in a 5-fold improvement in BER for the fish_3 image. The BER value of 1.688×10^{-1} improved for plants_5 image using QAM-64 at -10 dB and 1.42×10^{-1} improvement in BER for FSK. For fish_3 image sensitivity is 2.9×10^{-3} using QAM-64 and 23.4×10^{-3} using FSK. For plants_5 image the sensitivity is 5.2×10^{-3} using QAM-64 and 20.66×10^{-3} using FSK modulation.

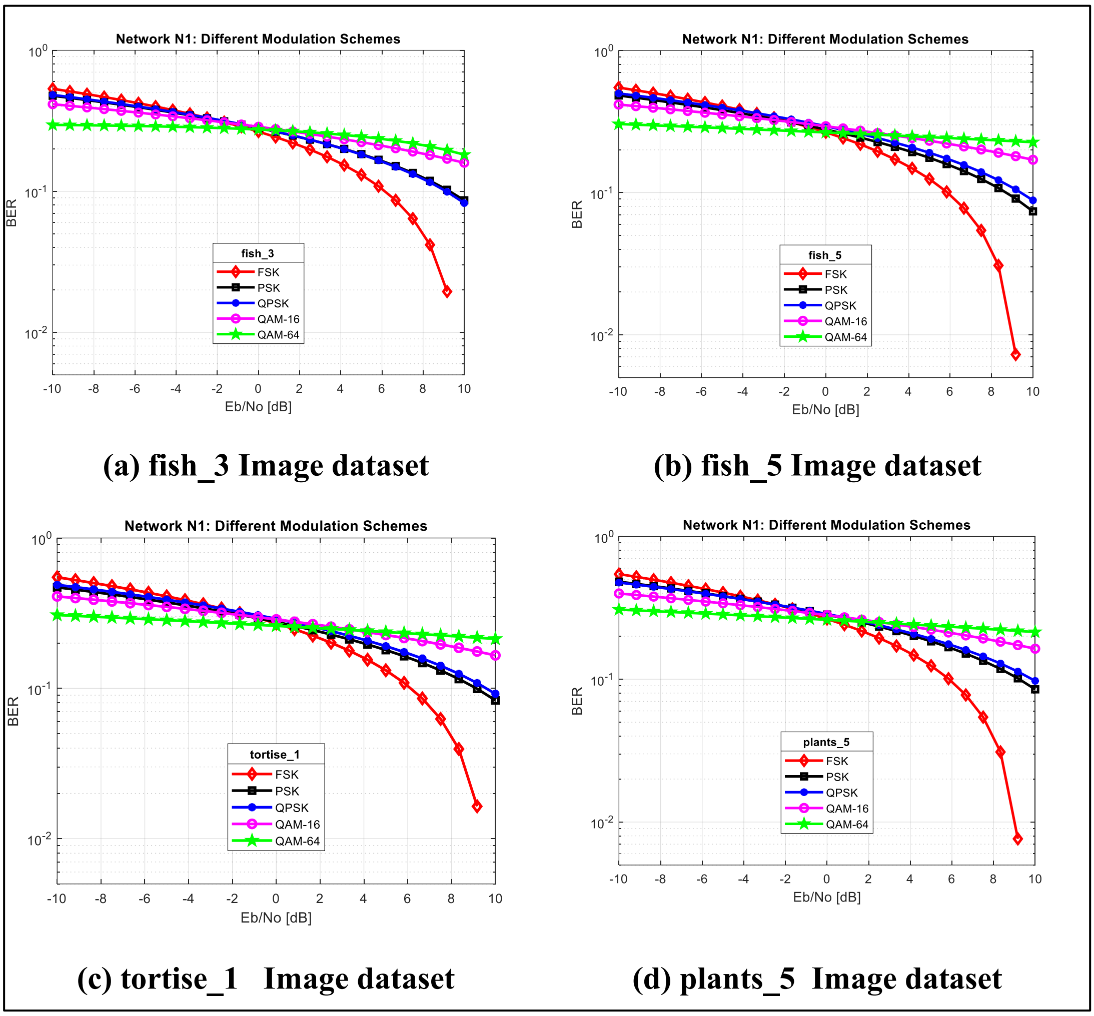
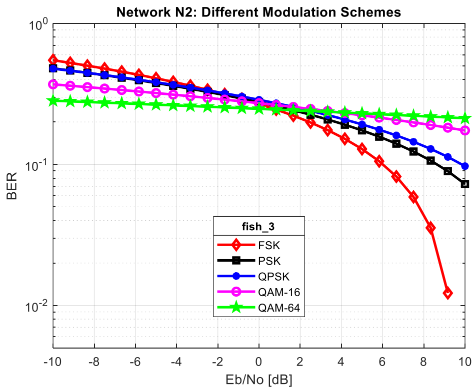


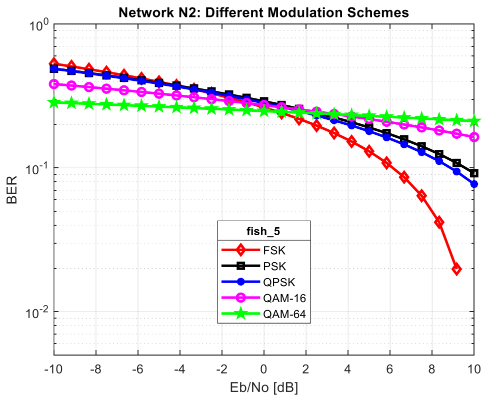
Figure 12. BER vs. Eb/N0 [dB] response for NN model with network configuration (16:4:16)

Table 4 presents a comparison of NN model metrics for image transmission with SNRs set to -5 dB and +5 dB. At -5 dB SNR, the 64:8:64 network model exhibited lower PSNR for all four image datasets, while the 16:4:16 NN model had a lower MSE. The entropy of the reconstructed image was lower and closer to the entropy of the input image. At +5 dB SNR, the MSE was lower for the 16:4:16 network model, and the entropy of the reconstructed images was lower than that of the 64:8:64 models. Considering the results in both cases, the 16:4:16 NN model is recommended. However, for a higher compression ratio, the 64:8:64 models are recommended with a BER trade-off.

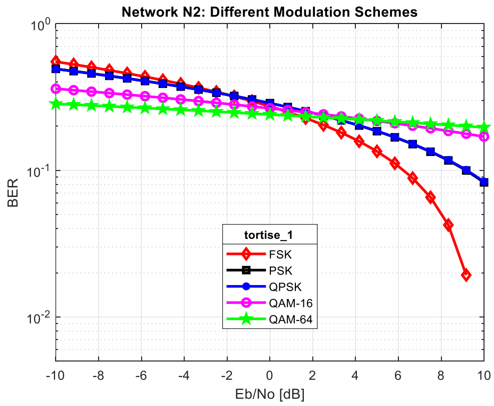
Box plots of PSNR, SSI, Entropy Out, and MSE are presented with SNR variations from -10 dB to 10 dB for one image selected from each of four groups for the 16:4:16 network model (Fig. 14). The Plants_5 image exhibits a higher entropy of 5.4413 and a comparatively lower PSNR of 16.8786 dB, whereas the fish_5 image demonstrates a PSNR of 20.4296 dB with an entropy value of 3.2246. For the fish_3 image in the group-1 dataset, as SNR increased from -10 dB to +10 dB, PSNR and SSI increased by 1.306 and 0.1945, respectively, indicating improvements in image quality and structural similarity. The MSE and BER decreased by 237.74 and 0.0107, respectively, showing reduced error. The output entropy also reduced by 0.7704. Similarly, for the fish_5 image from the group-2 dataset, the PSNR and SSI improved by 1.864 and 0.2005. MSE and output entropy reduced by 307.02 and 0.7729. For the tortoise_1 image belonging to the group-3 dataset, the PSNR and SSI increased by 1.454 and 0.1296. The MSE and output entropy values decreased by 345.5 and 0.429, respectively. For the plants_5 image corresponding to the group-4 dataset, the PSNR and SSI enhanced by 0.995 and 0.1496. The MSE and output entropy dropped by 338.5 and 0.0019. Across all images, an increase in SNR from -10 dB to +10 dB resulted in improved image quality, structural similarity, reduced error, and complexity. The output entropy of the reconstructed images decreases with a change in SNR from -10 dB to +10 dB, indicating less information loss and higher fidelity.



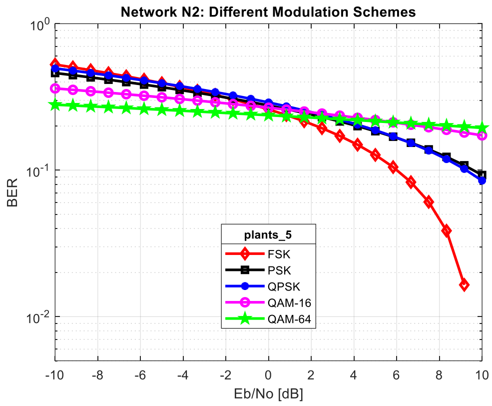
(a) fish_3 Image dataset



(b) fish_5 Image dataset



(c) tortise_1 Image dataset



(d) plants_5 Image dataset

Figure 13. BER vs. EB/N0 [dB] response for NN model with network configuration (64:8:64)

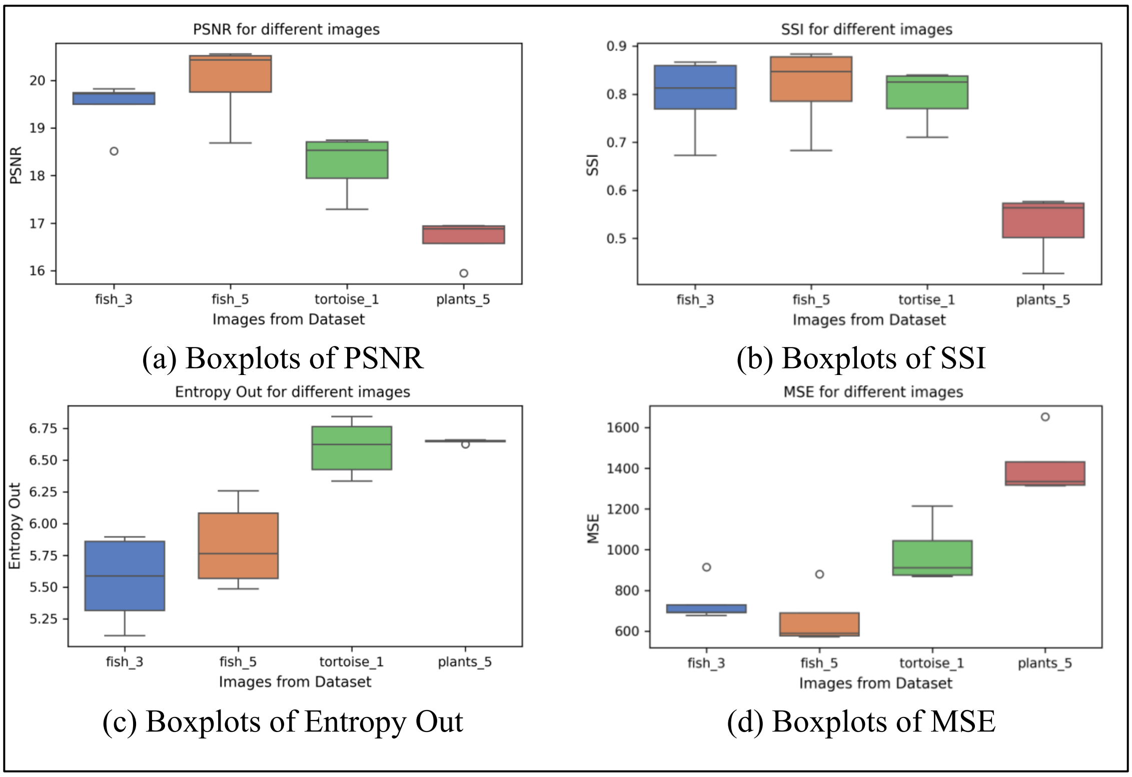
The four dataset images illustrate how SNR levels affect PSNR values. As SNR declined from 0 dB to -10 dB, PSNR values dropped from 19.7199 to 18.5164 for fish_3, from 20.4296 to 18.6876 for fish_5, from 18.5304 to 17.2896 for tortoise_1, and from 16.8786 to 15.9495 for plants_5. Conversely, raising SNR from 0 to +10 dB significantly improved PSNR: for fish_3 from 19.7199 to 19.8225, for fish_5 from 20.4296 to 20.5520, for tortoise_1 from 18.5304 to 18.7444, and for plants_5 from 16.8786 to 16.9453. This pattern suggests that greater SNR levels improve image quality, while lower SNR levels diminish it.

Reconstructed images after decoding at -10 dB and +10 dB SNR are shown in Figure 15. We observed that the reconstructed image had checkerboard patterns as it was grouped into sub-images, and the image quality was significantly degraded compared to the input image. To improve visual clarity and reconstruct the image with less distortion, a Neural Network-based image enhancement model is used.

Table 4. Performance comparison of NN models (16:4:16 and 64:8:64) at -5 dB and +5 dB SNR

Dataset Images	Entropy in	SNR	NN Models	PSNR	SSI	MSE (* $10^{3)}$	BER	Entropy Out
fish_3	2.4442	-5dB	16:4:16	19.5010	0.7692	0.7294	0.0371	5.8580
			64:8:64	13.0677	0.4043	3.208	0.0928	5.8136
		+5dB	16:4:16	19.7383	0.8593	0.690	0.0381	5.3159
			64:8:64	16.0697	0.5008	1.5921	0.0381	6.2736
fish_5	3.2246	-5dB	16:4:16	19.7515	0.7853	0.6885	0.0479	6.0806
			64:8:64	13.5239	0.4292	2.8886	0.0928	5.9303
		+5dB	16:4:16	20.5203	0.8773	0.576	0.0361	5.5686
			64:8:64	16.0723	0.5500	1.6064	0.0645	6.3082
tortoise_1	4.2670	-5dB	16:4:16	17.9457	0.7701	1.043	0.0449	6.8411
			64:8:64	11.4539	0.4167	4.2551	0.1748	6.6607
		+5dB	16:4:16	18.7063	0.8371	0.912	0.0332	6.4234
			64:8:64	14.1375	0.5237	2.5080	0.0859	6.9906
plants_5	5.4413	-5dB	16:4:16	16.5719	0.5015	1.4318	0.1416	6.6523
			64:8:64	12.7400	0.2026	3.4600	0.2393	6.6329
		+5dB	16:4:16	16.9352	0.5727	1.3169	0.1387	6.6537
			64:8:64	14.9554	0.2733	2.2092	0.1738	6.8250

The JPEG2000, SPIHT, Vector Quantization, and Huffman Coding results reported in Table 5 are adopted from the benchmark MATLAB-based evaluation presented in [3], where these standard algorithms were evaluated under identical experimental settings using the same compression ratio formulation. For the proposed lightweight DNN models, the compressed size corresponds to the DNN encoder latent representation, ensuring a fair and consistent comparison across all methods. After image enhancement, Proposed-1 achieved a PSNR of 28.98 and SSIM of 0.9235 for the same percentage of compression. In comparison, the reference method column in Table 5 presents a PSNR of approximately 18 dB, indicating an improvement of about 50% with respect to the proposed framework.



(a) Boxplots of PSNR (b) Boxplots of SSI

(c) Boxplots of Entropy Out (d) Boxplots of MSE

Figure 14. Box plots of PSNR, SSI, entropy out, and MSE for four different images in the dataset under varying SNR conditions

The semantic communication approach in [32] transmits underwater images using micro volume textual semantics and achieves robustness under low SNR conditions. However, this approach prioritizes semantic consistency and does not preserve pixel-level fidelity or report PSNR and SSI for compressed pixel images. In contrast, the proposed lightweight DNN-based framework maintains competitive PSNR and SSI even under severe underwater acoustic channel conditions, including SNR values as low as -10 dB.

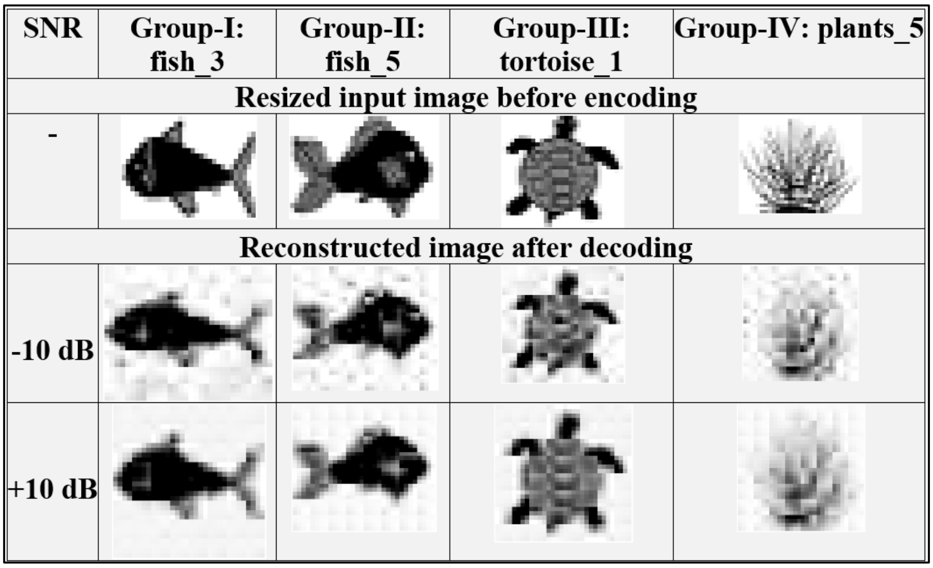


Figure 15. Reconstructed images after decoding under various SNR conditions

Table 5. Comparison of different compression algorithms performance

Parameters	JPEG 2000	SPIHT	Vector Quantization	Huffman Coding	Reference [3]	Proposed-1 16:4:16	Proposed-2 64:8:64	Proposed-3 256:16:256
Size	588 B	1119B	88B	671B	20B	256B	128B	64B
PSNR	29.65	24.49	19.24	21.72	18.19	20.5203	16.0723	14.9384
SSIM	0.54	0.66	0.90	0.40	0.88	0.8773	0.550	0.4935
Compression (%)	89.3	79.7	98.4	87.8	99.6	75	87.5	93.75

Table 5 represents the performance comparison of different compression algorithms. The compression percentage as defined in equation 15 is the relative reduction in data size compared to the original image.

$$\text{Compression (\%)} = \left(1 - \frac{\text{Compressed Size}}{\text{Orginal SIze}}\right) \text{x } 100 \tag{15}$$

Table 6. Performance comparison of different images before and after image enhancement

Image Data set	Entropy		Before Image Enhancement			After Image Enhancement		
		SNR	PSNR (dB)	SSI	Entropy	PSNR	SSI	Entropy
Group-I fish_3	2.4442	- 10 dB	18.5164	0.6724	5.8948	28.05	0.9372	2.2889
		+10 dB	19.8225	0.8669	5.1174	27.6881	0.9328	2.2627
Group-II fish_5	3.2246	- 10 dB	18.6876	0.6825	6.2568	27.98	0.9216	2.9622
		+10 dB	20.5520	0.8830	5.4839	28.98	0.9235	2.9914

Group-III tortoise_1	4.267	- 10 dB	17.2896	0.7099	6.7629	28.48	0.9620	3.9263
		+10 dB	18.7444	0.8395	6.3333	28.58	0.9620	3.9364
Group-IV plants_5	5.4413	- 10 dB	15.9495	0.4268	6.6256	27.6092	0.8868	5.0387
		+10 dB	16.9453	0.5764	6.6446	26.3204	0.8868	4.443

In this work, the baseline performance is defined as the reconstructed decoder image quality before applying the CNN–FC-based image enhancement model. Table 6 presents the results of the CNN-based image enhancement model for four test images under -10 and +10 dB SNR conditions. PSNR and SSI values were significantly increased after image enhancement for both noise levels. Under -10 dB SNR conditions, the baseline PSNR values for fish_3, fish_5, tortoise_1, and plants_5 were 18.5164, 18.6876, 17.2896, and 15.9495 dB respectively. After applying the proposed CNN-FC enhancement framework, the PSNR improved by approximately 50% across all tested images, Increasing from around 18 dB to nearly 28 dB, thereby demonstrating the effectiveness of the proposed method in enhancing underwater image quality under highly noisy conditions. After image enhancement, the SSI values were 0.9 for the tested image in the dataset, indicating better image visual qualities. For the range of -10 dB to +10 dB SNR channel conditions, enhanced image entropy values for fish_3 were2.2889 and 2.2627, nearly identical values to the input entropy of fish_3 werewas2.4442. Similar results were found for other images in the underwater image dataset, indicating retaining the image information. For comparison, the recent underwater acoustic semantic communication (UASC) approach by Zhang et al. [32] achieved semantic similarity scores (SSC) of approximately 0.944 under low SNR conditions (3–8 dB), demonstrating strong robustness in preserving key image features. The proposed framework achieves a comparable SSIM of approximately 0.93 at -10 dB SNR, confirming the effectiveness of the end-to-end compression, modulation, and channel-aware reconstruction pipeline for underwater image transmission.

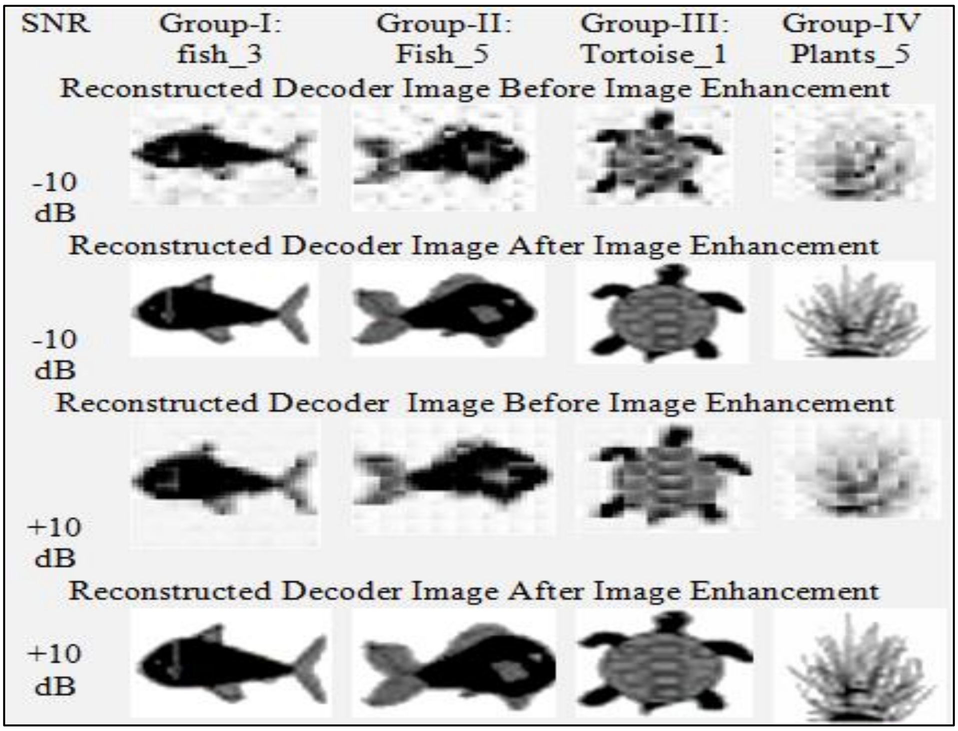


Figure 16. Reconstructed 16:4:16 NN model decoder images before and after deep learning based image enhancement under various SNR conditions

Figure 16 shows visual comparisons of different images from the dataset, displaying the reconstructed noisy input and enhanced images for -10 db and +10 dB SNR conditions. At -10 dB the image distortion was high and the Convolutional Neural Network-Fully Connected layers (CNN-FC) image enhancement model removed the noise and enhaced the image for better visual quality.

The proposed DNN based encoder-decoder framework was developed to address the key challenges of underwater image transmission systems. In such systems, attenuation, multipath propagation and scattering in the acoustic channel significantly degrade the transmitted image quality. In real-world underwater scenarios, such as marine ecology monitoring, autonomous underwater vehicle (AUV) operations and seabed mapping, transmitting reliable images efficiently through noisy, bandwidth-limited channels is essential.

The DNN encoder compresses images into a smaller latent representation, minimizing transmission bandwidth while preserving essential image features. This is suitable for bandwidth constrained and low power underwater communication systems. The encoder reduces redundancy through learned nonlinear mapping and the decoder reconstructs images despite channel distortions. Quantitative analysis using MSE, PSNR and SSI demonstrates high image reconstruction quality. PSNR improved from 20.52 dB to 28.98 dB and SSI increased from 0.8773 to 0.9235, indicating effective minimization of MSE and enhanced perceptual similarity. In the underwater imaging context, this yields sharper structural details with reduced blur, enabling more reliable visual inference for object detection and navigation.

By evaluating the system across multiple modulation schemes (FSK, PSK, QPSK, QAM-16, QAM-64) and SNR ranging from -10 dB to +10 dB, the proposed UWIT framework demonstrated consistent robustness under varying underwater acoustic noise conditions. QAM-64 exhibited stable BER behaviour with low sensitivity (3.4 × 10-3), highlighting its suitability for higher data rate transmission. Entropy-based dataset grouping enhanced generalization, enabling adaptation to low-visibility, low-contrast and clear underwater images.

The computational complexity of the proposed DNN architectures was analyzed using trainable parameter count and multiply-accumulate (MAC) operations, which represent memory requirements and processing complexity, respectively. For a fully connected layer, the parameter count was computed as $(n_{in} \times n_{out}) + n_{out}$, where n_{in} and n_{out} represent the number of input and output neurons. The computational complexity was determined by $n_{in} \times n_{out}$ MAC operations. Based on the architectural configurations the 16:4:16, 64:8:64, 256:16:256 and 64:32:8:32:64 architectures require 148, 1096, 8464 and 4744 trainable parameters with corresponding computational costs of 128, 1024, 8192 and 4608 MAC operations. Increasing network width and depth resulted in higher representational capacity at the expense of increased computational complexity. The 64:8:64 architecture maintains a balance between computational cost, compression efficiency and image reconstruction performance.

Efficient data transmission is an important requirement in underwater image communication systems, where channel impairments and limited acoustic bandwidth can significantly affect transmission reliability and data throughput [33]. The proposed framework addresses this challenge by compressing image data into compact latent feature representations prior to transmission. As reported in Table 5, the 16:4:16, 64:8:64, and 256:16:256 architectures achieve compression ratios of 75%, 87.5%, and 93.75%, respectively, corresponding to payload reductions of 4×, 8×, and 16× relative to the original representation. These reductions decrease the amount of transmitted data, thereby lowering bandwidth requirements and communication overhead. While the 256:16:256 architecture achieves the highest compression efficiency, the

64:8:64 architecture maintains a balance between data reduction and reconstructed image quality.

The proposed framework has potential applicability in Autonomous Underwater Vehicle (AUV) applications requiring bandwidth-efficient underwater image transmission. Recent studies emphasize the importance of communication-efficient perception and communication-aware autonomy in future underwater robotic systems [34], [35]. The proposed framework achieves up to 93.75% data compression while maintaining reconstruction quality. The reconstructed images attain a PSNR of 28.98 dB and an SSIM of 0.9235, demonstrating effective preservation of visual and structural information for underwater observation, target monitoring, environmental assessment, and navigation support. The framework therefore represents a promising solution for bandwidth-efficient visual data transmission in future autonomous underwater systems.

Although the proposed lightweight fully connected DNN architectures demonstrate favorable compression performance, practical deployment on autonomous underwater vehicles (AUVs) involves several challenges that extend beyond algorithmic accuracy. Real-time feasibility is determined not only by parameter count and MAC operations but also by processor throughput, memory bandwidth, cache behavior, task scheduling and power availability. During underwater missions, the onboard processor must simultaneously execute navigation, localization, obstacle avoidance, sensor fusion, acoustic communication and image processing under strict energy constraints. Consequently, reducing computational complexity decreases both arithmetic workload and memory usage which are major contributors to inference latency and energy consumption in embedded systems.

The proposed architectures require only 148 to 8464 trainable parameters and 128 to 8192 MAC operations, which are substantially lower than those of recent CNN based underwater image compression networks. In [36], 24.8 million parameters and 61.1 GFLOPs are used, while transformer based variants exceed 75 million parameters to maximize rate distortion performance. These observations indicate that contemporary underwater compression systems often achieve higher reconstruction fidelity at the cost of significantly increased computational and memory requirements. In contrast, the proposed lightweight fully connected architecture prioritizes embedded deployment and computational efficiency, representing a different engineering trade-off between compression performance and implementation complexity. Bandwidth reduction must also be interpreted in the context of underwater acoustic communication constraints. Underwater channels remain fundamentally limited by low available bandwidth, long propagation delay, severe multipath propagation, Doppler spread, attenuation and elevated bit error rates. Therefore, the achieved compression ratios of 75%, 87.5%, and 93.75% should be viewed as reductions in channel occupancy and communication overhead rather than complete solutions to underwater communication limitations. Recent underwater communication studies emphasize that practical systems must simultaneously balance compression efficiency, transmission robustness, and reconstruction fidelity because aggressive compression can increase sensitivity to channel impairments during long range acoustic transmission [33], [37].

Another important deployment consideration is robustness under non ideal channel conditions. Experimental studies on underwater acoustic image transmission have shown that burst errors, fading, and intermittent packet loss can severely degrade conventional image codecs, even when compression performance is satisfactory under ideal conditions. Consequently, operational underwater systems often require additional mechanisms such as channel aware coding, adaptive retransmission, unequal error protection, or joint source-channel optimization to maintain recognizable visual information under high BER conditions [37]. The present work focuses on compression efficiency under simulated channel conditions therefore, robustness enhancement under realistic underwater acoustic impairments remains an important direction for future investigation.

Finally, the current MATLAB based evaluation demonstrates algorithmic feasibility rather than complete hardware level real-time feasibility. Practical deployment requires implementation on FPGA, embedded GPU or edge-AI platforms to measure processor utilization, memory bandwidth, execution latency, throughput and power consumption under operational conditions. Hardware experiments and sea trial validation are therefore necessary before the framework can be considered fully operational for long duration AUV missions. Nevertheless, the extremely low parameter count and MAC complexity of the proposed architecture suggest that it is substantially more suitable for resource constrained underwater platforms than many contemporary CNN and transformer based compression networks.

5. Conclusion

This work demonstrates a lightweight DNN-based framework for bandwidth efficient underwater image transmission by jointly addressing image compression, channel transmission and image reconstruction. The proposed encoder-decoder framework establishes a trade-off among compression efficiency, computational complexity and reconstructed image quality. The evaluated architectures achieve compression ratios of 75% to 93.75% with only 148 to 8,464 trainable parameters and 128 to 8,192 multiply accumulate (MAC) operations, demonstrating low computational complexity. The results demonstrate a compression reconstruction trade-off across the DNN architectures, with the 64:8:64 configuration providing an intermediate operating point between compression efficiency and reconstructed image quality. Transmission performance is jointly influenced by the DNN architecture, modulation scheme and channel SNR over the evaluated range of -10 to +10 dB. QAM-64 exhibited low sensitivity to SNR variation, whereas FSK achieved lower BER under selected operating conditions. The CNN-FC image enhancement stage substantially improved the quality of the reconstructed images, increasing PSNR from 20.5203 to 28.98 dB and SSI from 0.8773 to 0.9235. At an SNR of -10 dB, the fish_3 image exhibited improvements of 51.48% in PSNR and 39.38% in SSI demonstrating effective recovery under severe channel degradation. Overall, the results support the use of lightweight DNN-based compression for bandwidth-constrained underwater image transmission. Future work will focus on real-channel validation and adaptive, channel-aware DNN models for dynamic compression and reconstruction under time-varying SNR, fading, and Doppler effects, followed by implementation on embedded edge-computing platforms for real-time underwater sensing and AUV applications.

References

[1] Junejo, Naveed Ur Rehman, Hamada Esmaiel, Mingzhang Zhou, Haixin Sun, Jie Qi, and Junfeng Wang. "Sparse Channel Estimation of Underwater TDS-OFDM System Using Look-Ahead Backtracking Orthogonal Matching Pursuit." IEEE Access 6 (2018): 74389-74399.

[2] Jing, Lianyou, Qingsong Wang, Chengbing He, and Xuewei Zhang. "A Learned Denoising-Based Sparse Adaptive Channel Estimation for OTFS Underwater Acoustic Communications." IEEE Wireless Communications Letters 13, no. 4 (2024): 969-973.

[3] Danckaers, Alice, and Mae L. Seto. "Transmission of Images by Unmanned Underwater Vehicles." Autonomous Robots 44, no. 1 (2020): 3-24.

[4] Haziq, Muhammad, Quoc Viet Phung, Stefan Lachowicz, Daryoush Habibi, and Iftekhar Ahmad. "Modulation Techniques for Underwater Acoustic Communication: A Comprehensive Survey." IEEE Access (2025). 150715– 150755.

[5] Gangisetty, Shankar, and Raghu Raj Rai. "Underwater Image Restoration Using Deep Encoder–Decoder Network with Symmetric Skip Connections." Signal, Image and Video Processing 16, no. 1 (2022): 247-255.

[6] Naval Mine Warfare. 2001. “Operational and Technical Challenges for Naval Forces”.National Academies Press EBooks. Washington, DC https://doi.org/10.17226/10176.

[7] Sahu, Ghanshyam, and Virendra Kumar Swarnkar. "Advancing Underwater Image Quality Enhancement Through Hybrid Deep Learning Architectures." In International Conference on Artificial Intelligence on Textile and Apparel, Singapore: Springer Nature Singapore, 2024, 767-778.

[8] Zhang, Yunfeng, Qun Jiang, Peide Liu, Shanshan Gao, Xiao Pan, and Caiming Zhang. "Underwater Image Enhancement Using Deep Transfer Learning Based on a Color Restoration Model." IEEE Journal of Oceanic Engineering 48, no. 2 (2023): 489-514.

[9] Dai, Chenggang, and Mingxing Lin. "Single Underwater Image Restoration Using Variational Framework Guided by Imaging Model with Noise." IEEE Access 12 (2024): 82427-82442.

[10] R. Su, Z. Gong, H. Cheng and C. Li, "Fast Underwater Target Localization with Wideband Signals for the Internet-of-Underwater-Things," in IEEE Internet of Things Journal, vol. 12, no. 15, 1 Aug.1, 2025, 31561-31574.

[11] Domingo, Mari Carmen. "Overview of Channel Models for Underwater Wireless Communication Networks." Physical Communication 1, no. 3 (2008): 163-182.

[12] Kerai, Shivani, and Ganesh Khekare. "Contextual Embedding Generation of Underwater Images Using Deep Learning Techniques." IAES Int J Artif Intell (IJ-AI) 13, no. 3 (2024): 3111-3118.

[13] Chang, Y. "A Comprehensive Review of Underwater Optical, Acoustic, and Electromagnetic Communication." Applied and Computational Engineering 145, no. 1 (2025): 197-202.

[14] Li, Yuhan, Ruizhi Ruan, Zupeng Zhou, Anqing Sun, and Xiaonan Luo. "Positioning of Unmanned Underwater Vehicle Based on Autonomous Tracking Buoy." Sensors 23, no. 9 (2023): 4398.

[15] Chen, Zhipeng, Zhiqiang He, Kai Niu, and Yue Rong. "Neural Network-Based Symbol Detection in High-Speed OFDM Underwater Acoustic Communication." In 2018 10th

International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2018, 1-5.

[16] van Walree, Paul A., François-Xavier Socheleau, Roald Otnes, and Trond Jenserud. "The Watermark Benchmark for Underwater Acoustic Modulation Schemes." IEEE journal of oceanic engineering 42, no. 4 (2017): 1007-1018.

[17] Stojanovic, Milica, and James Preisig. "Underwater Acoustic Communication Channels: Propagation Models and Statistical Characterization." IEEE communications magazine 47, no. 1 (2009): 84-89.

[18] Zheng, Shijian, Rujing Wang, Shitao Zheng, Fenmei Wang, Liusan Wang, and Zhigui Liu. "A Multi-Scale Feature Modulation Network for Efficient Underwater Image Enhancement." Journal of King Saud University Computer and Information Sciences 36, no. 1 (2023): 101888.

[19] Chen, Weiqi, Hua Yu, Quansheng Guan, Fei Ji, and Fangjiong Chen. "Reliable and Opportunistic Transmissions for Underwater Acoustic Networks." IEEE Network 32, no. 4 (2018): 94-99.

[20] Jing, Lianyou, Qingsong Wang, Chengbing He, and Xuewei Zhang. "A Learned Denoising-Based Sparse Adaptive Channel Estimation for OTFS Underwater Acoustic Communications." IEEE Wireless Communications Letters 13, no. 4 (2024): 969-973.

[21] Felamban, Muhamad, Basem Shihada, and Kamran Jamshaid. "Optimal Node Placement in Underwater Wireless Sensor Networks." In 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2013, 492-499.

[22] Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. Deep Learning. Vol. 1, no. 2. Cambridge: MIT press, 2016.

[23] O'shea, Timothy, and Jakob Hoydis. "An Introduction to Deep Learning for the Physical Layer." IEEE Transactions on Cognitive Communications and Networking 3, no. 4 (2017): 563-575.

[24] Adil, Muhammad, Songzuo Liu, Suleman Mazhar, Ayman Alharbi, Honglu Yan, and Muhammad Muzzammil. "A Novel Transfer Learning-Based OFDM Receiver Design for Enhanced Underwater Acoustic Communication." Journal of Marine Science and Engineering 13, no. 7 (2025): 1284.

[25] Zhu, Zhengliang, Yuehai Zhou, Rong Wang, and Feng Tong. "Internet of Underwater Things Infrastructure: A Shared Underwater Acoustic Communication Layer Scheme for Real-World Underwater Acoustic Experiments." IEEE Transactions on Aerospace and Electronic Systems 59, no. 5 (2023): 6991-7003.

[26] Liu, Lei, Lin Cai, Lu Ma, and Gang Qiao. "Channel State Information Prediction for Adaptive Underwater Acoustic Downlink OFDMA System: Deep Neural Networks Based Approach." IEEE Transactions on Vehicular Technology 70, no. 9 (2021): 9063-9076.

[27] Yang, Hongkun, Bin Liu, Fengyuan Ren, Hao Wen, and Chuang Lin. "Optimization of Energy Efficient Transmission in Underwater Sensor Networks." In GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, IEEE, 2009, 1-6.

[28] Hu, Kuo-Jui, Yi-Tsung Pan, Li-Wei Jiang, Sin-Der Lee, and Sheng-Long Kao. "A Robust Underwater Image Enhancement Algorithm." The Journal of Supercomputing 81, no. 1 (2025): 244.

[29] Stojanovic, Milica. "On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel." ACM SIGMOBILE Mobile Computing and Communications Review 11, no. 4 (2007): 34-43.

[30] Onasami, Oluwaseyi, Damilola Adesina, and Lijun Qian. "Underwater Acoustic Communication Channel Modeling Using Deep Learning." In Proceedings of the 15th International Conference on Underwater Networks & Systems, 2021, 1-8.

[31] Zhang, Youwen, Junxuan Li, Yuriy Zakharov, Xiang Li, and Jianghui Li. "Deep Learning Based Underwater Acoustic OFDM Communications." Applied Acoustics 154 (2019): 53-58.

[32] Zhang, Ying, Huanyu Li, Bingyu Li, Li Li, Weibo Zhang, Hao Wang, and Peng Ren. "An Underwater Acoustic Semantic Communication Approach to Underwater Image Transmission: Y. Zhang et al." Intelligent Marine Technology and Systems 3, no. 1 (2025): 5.

[33] Busacca, Fabio, Laura Galluccio, Sergio Palazzo, Andrea Panebianco, Zhuoran Qi, and Dario Pompili. "Adaptive Versus Predictive Techniques in Underwater Acoustic Communication Networks." Computer Networks 252 (2024): 110679.

[34] Wibisono, Arif, Md Jalil Piran, Hyoung-Kyu Song, and Byung Moo Lee. "A Survey on Unmanned Underwater Vehicles: Challenges, Enabling Technologies, and Future Research Directions." Sensors 23, no. 17 (2023): 7321.

[35] Cao, Wenqiang, Jing Yan, Xian Yang, Cailian Chen, and Xinping Guan. "Communication-Aware Mobile Relaying via an AUV for Minimal Wait Time: A Broad Learning-Based Solution." IEEE/CAA Journal of Automatica Sinica 11, no. 3 (2024): 797-799.

[36] Li, Mengyao, Liquan Shen, Xia Hua, and Zhaoyi Tian. "EUICN: An Efficient Underwater Image Compression Network." IEEE Transactions on Circuits and Systems for Video Technology 34, no. 7 (2024): 6474-6488.

[37] Liu, Jiahui, Fei Yuan, Chang Xue, Zhenyu Jia, and En Cheng. "An Efficient and Robust Underwater Image Compression Scheme Based on Autoencoder." IEEE Journal of Oceanic Engineering 48, no. 3 (2023): 925-945.