Journal of Advanced Zoology
ISSN: 0253-7214
Volume 38 Issue 1 Year 2017 Page 14-24

UNDERWATER IMAGE ENHANCEMENT USING A REDUCED-DEPTH U-NET: *Design, Implementation, and Evaluation*

Pankaj Pateriya[1*], Vijay Choudhary[1]

[1]*Department of Computer Science and Engineering, IPS Academy, Institute of Engineering & Science, Indore, Madhya Pradesh, India*
**Corresponding Author: Pankaj Pateriya, ppateriya@gmail.com, vij.choudhary@gmail.com*

Abstract

Background: Underwater images are commonly affected by wavelength-dependent light absorption and scattering, which produce colour distortion, reduced contrast and haze-like degradation. The aim of this study was to design, implement and evaluate a supervised deep-learning system for underwater image enhancement based on a reduced-depth U-Net encoder–decoder architecture.

Methods: The network was trained end-to-end on paired underwater imagery from the EUVP (Enhancing Underwater Visual Perception) dataset, obtained through a Kaggle mirror, using an L1 reconstruction loss and the Adam optimizer, without data augmentation, batch normalization, dropout, residual connections or adversarial training. A fixed-seed 80/10/10 split produced 9,149 training, 1,143 validation and 1,143 test pairs. Performance on the held-out test split was measured with two full-reference metrics, Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM), and with a custom no-reference metric adapted from the Underwater Image Quality Measure (UIQM).

Results: The system achieved an average PSNR of 29.36 dB, an average SSIM of 0.8610 and an average custom UIQM-style score of 3.5692 on the 1,143-pair test split. The trained model was additionally deployed through an interactive Streamlit application supporting image upload, adjustable-resolution inference, before/after comparison and no-reference quality feedback.

Conclusions: A transparent, purely supervised reduced-depth U-Net pipeline, with a deterministically defined dataset organization, quantitative evaluation and an accessible interactive interface, provides a practical and reproducible basis for underwater image enhancement and for further development.

Keywords *underwater image enhancement; U-Net; deep learning; image-to-image translation; supervised learning; PSNR; SSIM; UIQM; Streamlit deployment.*

1. Introduction

Images captured underwater undergo physical degradation processes with no direct counterpart in ordinary open-air photography. As light propagates through water, it is absorbed and scattered in a wavelength-dependent manner, with longer wavelengths attenuating substantially faster than shorter wavelengths. [1] This differential attenuation produces the blue-green colour cast characteristic of underwater photography, while scattering from suspended particles reduces contrast and introduces haze-like veiling. These effects degrade underwater imagery for both human interpretation and downstream computer-vision tasks, motivating decades of research into restoration and enhancement techniques tailored to the underwater domain.

Two broad families of approaches address underwater image degradation. The first comprises traditional, non-learning restoration and enhancement methods, including physically motivated dehazing priors and multiscale image-fusion techniques. The second, more recent family comprises data-driven deep-learning approaches that learn a mapping from degraded to enhanced images directly from training data, without an explicit physical model of image formation.

This paper documents a system belonging to the second family: a supervised, purely convolutional U-Net-style encoder–decoder network trained end-to-end to map underwater input images to enhanced reference images, using paired data from the EUVP dataset. The system additionally comprises an inference pipeline, a set of quantitative evaluation scripts and an interactive Streamlit-based demonstration application.

The contribution of this work is deliberately scoped. It does not introduce a new architecture, dataset or evaluation metric. Instead, it contributes: (1) a transparent, purely supervised implementation of the U-Net encoder–decoder pattern applied to underwater image enhancement; (2) a clearly documented paired-image enhancement pipeline spanning preprocessing, training and inference; (3) dataset organization, training and evaluation procedures accompanied by a dataset split that is reproducible through a fixed random seed; (4) an interactive deployment, implemented as a Streamlit application, that makes the trained model directly usable; and (5) a contextual, literature-based quantitative comparison against representative pre-2018 underwater enhancement methods, using representative reported values rather than independently reproduced experiments. The remainder of this paper reviews related work, describes the materials and methods, presents the results, describes the deployed application and discusses limitations and future directions.

2. Related Work

2.1 Physical modelling of underwater image formation

The classical computational treatment of underwater image formation was established by Jaffe, who modelled an underwater image as a superposition of direct, forward-scattered and backscattered light components grounded in the inherent and apparent optical properties of water. [1] This physical framing implicitly underlies essentially all subsequent underwater restoration work, whether physics-based or, as in this study, purely data-driven.

2.2 Prior-based restoration and the underwater dark channel prior lineage

The Dark Channel Prior (DCP) introduced by He, Sun and Tang provided an influential statistical prior for single-image haze removal in general (non-underwater) scenes, exploiting the observation that haze-free outdoor patches typically contain low-intensity pixels in at least one colour channel. [3] Because the standard DCP is unreliable underwater due to rapid attenuation of the red channel, Drews et al. proposed the Underwater Dark Channel Prior (UDCP), restricting the dark-channel computation to the blue and green channels, and an extended journal treatment incorporating joint depth estimation followed. [6-7] Together, these works constitute a coherent prior-based restoration lineage requiring no training data.

2.3 Colour correction and fusion-based enhancement

Chiang and Chen combined DCP-based dehazing with an explicit wavelength-compensation step accounting for differential attenuation of red, green and blue light underwater. [4] Ancuti et al. proposed a multiscale fusion approach blending a white-balanced image with a contrast-enhanced derivative via Laplacian pyramid weight maps, without requiring scene depth or specialized hardware. [5] Both methods remain widely used baselines in the underwater enhancement literature and, together with the UDCP lineage, demonstrate that competitive underwater enhancement was achieved without training data prior to the widespread adoption of deep learning in this domain.

2.4 Early CNN-based restoration

Outside the underwater-specific literature, Cai et al. introduced DehazeNet, a convolutional network trained to estimate the medium transmission map directly from a hazy image, replacing a hand-crafted component of the atmospheric scattering model with a learned one. [8] Although not designed for underwater imagery, DehazeNet is representative of the broader shift toward replacing hand-crafted restoration priors with learned components, a shift that the end-to-end U-Net of the present work continues without retaining any explicit physical scattering model.

2.5 Encoder–decoder architectures and image-to-image translation

The U-Net architecture, introduced by Ronneberger, Fischer and Brox for biomedical image segmentation, established the symmetric encoder–decoder pattern with concatenation-based skip connections that the network in this work directly instantiates, at reduced depth and for a regression rather than a segmentation task. [9] Isola et al. subsequently showed that a U-Net-style generator, trained with a combined adversarial and L1 objective within a conditional GAN framework (pix2pix), could perform general-purpose paired image-to-image translation. [12] The network used in this work is architecturally analogous to the pix2pix generator alone, trained with an L1 objective only, without any adversarial or discriminator component.

2.6 Underwater-specific quality metrics

Yang and Sowmya proposed the Underwater Colour Image Quality Evaluation (UCIQE) metric, a linear combination of chroma, saturation and contrast statistics computed in CIELab colour space. [10] Panetta, Gao and Agaian proposed the Underwater Image Quality Measure (UIQM), a human-visual-system-inspired composite of colourfulness (UICM), sharpness (UISM) and contrast (UIConM) sub-measures. [11] UIQM is the metric family adapted, in simplified form, by the evaluation pipeline of this work (see Evaluation metrics and Results), whereas UCIQE is not implemented and is discussed here only as context. Wang et al. introduced the Structural Similarity Index (SSIM), a general-purpose full-reference metric used directly in the evaluation pipeline of this work. [2]

2.7 Research gap

The reviewed literature consists primarily of (a) traditional, non-learning restoration and fusion methods requiring no training data, grounded in prior-based dehazing adapted from general computer vision, and (b) a general-purpose encoder–decoder architecture and its extension to conditional adversarial image-to-image translation, neither originally demonstrated on underwater imagery. The contribution of this work is centred on the practical adaptation, systematic implementation and evaluation of a reduced-depth U-Net for paired underwater image enhancement: a transparently implemented, purely supervised, non-adversarial application of the U-Net encoder–decoder pattern, evaluated with a general-purpose full-reference metric and an underwater-specific no-reference metric in the tradition of UIQM, with the full pipeline documented and paired with an interactive deployment interface.

3. Materials and Methods

The system comprises six functional components, implemented as independent scripts: a dataset-splitting utility; a training script that fits the U-Net to the training data while monitoring validation loss; an inference script that applies the trained model to a directory of test inputs; two evaluation scripts computing PSNR/SSIM and the custom UIQM-style metric, respectively; and a Streamlit application performing single-image inference with quality-metric feedback. Each stage consumes the output of the previous stage from a fixed default directory location (Figure 1).

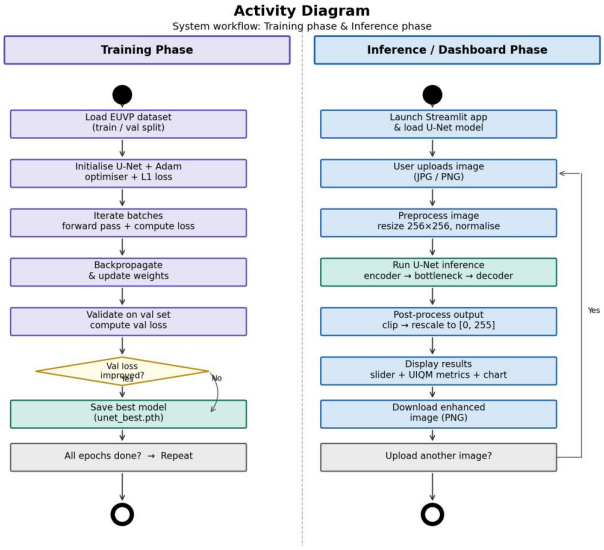

Figure 1. Overall workflow of the proposed underwater image enhancement framework, from dataset splitting through training, inference, evaluation and Streamlit deployment.

3.1 Dataset

The system uses the EUVP (Enhancing Underwater Visual Perception) dataset, obtained through a Kaggle mirror referenced in the project documentation. The dataset is organized as a paired image-to-image collection in which each degraded input image has a corresponding enhanced reference image of the same filename, with inputs and references stored in separate directories. The specific EUVP subset(s) used to construct the project data directory are not established by the available project documentation and are not asserted here.

3.2 Data preparation

The dataset is organized under a common root directory containing train, validation and test partitions, each with separate input and target sub-directories. A dedicated script partitions the original pool of paired files: filenames are collected from the original input directory, shuffled using a fixed random seed (seed = 42), and assigned such that the first 10% form the validation set, the next 10% form the test set and the remaining 80% form the training set. Matching input and reference files are moved together to preserve filename-based pairing. The split operates at the level of individual image files rather than scenes or capture sequences; no scene-aware separation or explicit duplicate detection across splits is implemented. Table 1 reports the verified counts obtained by directly inspecting the populated data directories.

Table 1. Dataset split (EUVP paired images)

Split	Input images	Target images	Pairs	Proportion
Training	9,149	9,149	9,149	80.0%
Validation	1,143	1,143	1,143	10.0%
Testing	1,143	1,143	1,143	10.0%
Total	11,435	11,435	11,435	100%

Split produced with a fixed random seed of 42; input and target files are paired by filename.

Filename matching between input and reference directories was verified within each split, confirming a corresponding pair for every observed sample. The resulting proportions are consistent with the intended 80/10/10 design of the splitting procedure.

3.3 Image preprocessing

An identical, deterministic preprocessing sequence is applied to every image at both training and inference time. Images are loaded with OpenCV (BGR channel order) and converted to RGB. Each image is resized to a fixed resolution of 256 × 256 pixels. Pixel intensities in [0, 255] are divided by 255.0 to obtain values in approximately [0, 1]. The resulting array is converted to a PyTorch tensor and permuted from height–width–channel (HWC) to channel–height–width (CHW) order. No additional preprocessing (mean/standard-

deviation normalization, histogram equalization, white-balance correction or denoising) is applied at any stage. No data augmentation (flipping, rotation, cropping, colour jitter or noise injection) is implemented in the training pipeline; every training sample is a deterministic function of the corresponding raw input image.

3.4 U-Net architecture

The network is a reduced-depth U-Net-style encoder–decoder with three encoder levels, a bottleneck and three decoder levels, in contrast to the four-level design of the original U-Net (Figure 2). [9] The basic building block, DoubleConv, consists of two consecutive 3 × 3 convolutions (padding = 1), each followed by a ReLU activation; no batch normalization, dropout or additive residual connection is present inside the block.

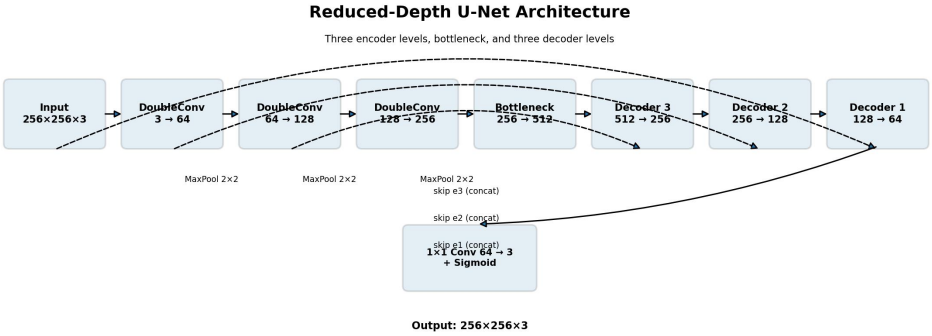

Figure 2. Architecture of the reduced-depth U-Net used in the proposed framework, showing channel progression and concatenation-based skip connections.

The encoder applies three DoubleConv stages with channel progression 3 → 64, 64 → 128 and 128 → 256, with 2 × 2 max-pooling between stages. The bottleneck applies a DoubleConv stage of 256 → 512. Each decoder level applies a 2 × 2 transposed convolution for upsampling, concatenates the result channel-wise with the corresponding encoder feature map and passes the concatenation through a DoubleConv stage that restores the target channel count. The output head applies a 1 × 1 convolution (64 → 3 channels) followed by a sigmoid activation, constraining predictions to approximately [0, 1]. Table 2 summarizes the architecture.

Table 2. U-Net architecture summary

Stage	Operation	In → Out channels
Encoder 1	DoubleConv	3 → 64
—	MaxPool2d (2 × 2)	64 → 64
Encoder 2	DoubleConv	64 → 128
—	MaxPool2d (2 × 2)	128 → 128
Encoder 3	DoubleConv	128 → 256
—	MaxPool2d (2 × 2)	256 → 256
Bottleneck	DoubleConv	256 → 512
Decoder 3	ConvT + concat(e3) + DoubleConv	512 → 256
Decoder 2	ConvT + concat(e2) + DoubleConv	256 → 128
Decoder 1	ConvT + concat(e1) + DoubleConv	128 → 64
Output	Conv2d (1 × 1) + Sigmoid	64 → 3

DoubleConv: two 3 × 3 convolutions (padding = 1), each followed by ReLU. ConvT: 2 × 2 transposed convolution; e1–e3: encoder feature maps.

All three skip connections are implemented as channel-wise concatenation rather than additive residual connections. This architecture is not presented as a novel contribution: it applies the published U-Net encoder–decoder pattern, at reduced depth, to an image-enhancement (regression) task rather than the original segmentation task, and without the original data-augmentation strategy. [9]

3.5 Training strategy

The model is implemented in PyTorch and trained end-to-end with an L1 (mean absolute error) loss between predicted and reference pixel values:

$$L_1 = (1/N)\ \Sigma_{i=1}^{N} |\hat{y}_i - y_i| \quad (1)$$

where $\hat{y}_i$ and y_i denote predicted and reference pixel values and N is the number of pixels. This is the only loss term used; no perceptual, SSIM-based, adversarial or colour-consistency component is included. Optimization uses Adam with a fixed learning rate of 0.0002 and default hyperparameters otherwise; no learning-rate scheduler or explicit weight decay is used. Training uses a batch size of 8 for a fixed 5 epochs, with no early-stopping criterion.

After each epoch, the model is evaluated on the validation set under torch.no_grad(), and the mean L1 loss across validation batches is computed as the sole validation signal. When the validation loss of the current epoch improves upon the best value observed so far, the model state dictionary is checkpointed (unet_best.pth); checkpoints are not saved at every epoch. Model selection for inference and evaluation is therefore based on best validation-loss checkpointing.

The dataset split is deterministic and controlled by a fixed random seed (see Data preparation), but the training script does not set an explicit PyTorch or NumPy random seed. Consequently, while the composition of the training, validation and test sets is reproducible, exact training-time reproducibility (weight initialization and batch shuffling) is not guaranteed by the provided code alone.

3.6 Inference pipeline

The inference procedure selects a CUDA device when available, falling back to CPU otherwise, instantiates the U-Net and loads the best-validation-loss checkpoint. For each test image, the preprocessing sequence described under Image preprocessing is applied, followed by a forward pass under torch.no_grad(). The network output is converted back to HWC layout, rescaled from [0, 1] to [0, 255], cast to 8-bit unsigned integer format, converted from RGB to BGR and written to disk using the same filename as the corresponding input, in a default output directory of ../results/images.

3.7 Experimental setup

All experiments use the split described under Data preparation: 9,149 training pairs, 1,143 validation pairs and 1,143 test pairs (11,435 pairs in total), produced with a fixed random seed of 42. Table 3 summarizes the training configuration used throughout the experiments. Images are processed at a fixed resolution of 256 × 256 pixels during both training and evaluation. Specific training hardware (GPU model, memory or wall-clock training time) is not recorded in the available project documentation and is not reported.

Table 3. Training configuration

Parameter	Value
Framework	PyTorch
Loss function	L1 (MAE)
Optimizer	Adam
Learning rate	0.0002
Learning-rate schedule	None
Batch size	8
Epochs	5
Input resolution	256 × 256
Checkpoint criterion	Best validation loss
Data augmentation	None

3.8 Evaluation metrics

Three quantitative metrics are implemented for distinct purposes: two full-reference metrics (PSNR, SSIM) comparing an enhanced output against its corresponding reference image, and one no-reference metric (the custom UIQM-style measure) that does not require a reference and can therefore also be applied to images without ground truth, such as user-uploaded images.

Peak Signal-to-Noise Ratio (PSNR) is computed as

$$PSNR = 20 \cdot \log_{10}(255 / \sqrt{MSE}) \quad (2)$$

where MSE is the mean squared error computed on pixel values in the 0–255 range between the enhanced output and its corresponding reference image, matched by filename and resized to 256 × 256 prior to comparison.

The Structural Similarity Index (SSIM) is computed via the structural_similarity function from scikit-image, using channel_axis = −1, data_range = 255 and win_size = 7. [2] In its general form, SSIM combines luminance, contrast and structure comparisons between local windows of two images:

$$SSIM(x, y) = [(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)] / [(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)] \quad (3)$$

where μ and σ denote local means and (co)variances and C_1 and C_2 are stabilizing constants.

A no-reference metric inspired by UIQM is implemented with simplified sub-components. [11] The colourfulness measure (UICM) is computed from the mean and standard deviation of the chrominance opponent channels rg = R − G and yb = 0.5(R + G) − B, rather than the standard asymmetric alpha-trimmed statistics. The sharpness measure (UISM) is the global mean of the Sobel gradient magnitude, rather than the standard block-based Enhancement Measure Estimation. The contrast measure (UIConM) is the plain global standard deviation of the grayscale image, rather than the standard block-based logarithmic contrast measure (AMEE). These are combined as

$$UIQM_s = (0.0282 \cdot UICM + 0.2953 \cdot UISM + 3.5753 \cdot UIConM) / 60 \quad (4)$$

where the division by 60 is a project-specific rescaling not present in the original published formulation. Because of these simplifications, values produced by this implementation are not directly comparable to canonical UIQM values reported elsewhere, and any such comparison should be treated with caution.

4. Results

Table 4 reports the aggregate results obtained by running the evaluation scripts over the held-out test split of 1,143 image pairs, using outputs produced by the inference pipeline of the trained model. The network achieved an average PSNR of 29.36 dB, an average SSIM of 0.8610 and an average custom UIQM-style score of 3.5692.

Table 4. Quantitative evaluation results on the test split (n = 1,143 pairs)

Metric	Type	Average value
PSNR	Full-reference	29.36 dB
SSIM	Full-reference	0.8610
Custom UIQM-style	No-reference	3.5692

PSNR and SSIM compare enhanced outputs with test-set reference images; the UIQM-style score is computed on the same enhanced outputs with the adapted implementation.

The UIQM-style score of 3.5692 should be interpreted specifically as an output of this adapted implementation, not as a value directly comparable to canonical UIQM figures reported in other publications. No per-image breakdown, standard deviation, confidence interval or minimum/maximum value is available from the evaluation scripts, and none is reported here. No baseline or comparison method was independently re-implemented under the same experimental conditions in this study. Table 5 instead situates these results against representative, literature-reported values for pre-2018 methods, and the comparison is discussed further in the Discussion.

Table 5. Contextual comparison with representative pre-2018 literature

Method	Method type	Year	PSNR (dB) ↑	SSIM ↑
UDCP [6]	Physical prior	2013 / 2016	~18.50	~0.7100
Fusion-based [5]	White-balance / contrast fusion	2012	~19.20	~0.7400
DehazeNet [8]	Early CNN restoration	2016	~21.10	~0.7800
Pix2Pix [12]	Conditional GAN	2017	~25.80	~0.8200
Proposed reduced U-Net	**Supervised encoder–decoder**	**—**	**29.36**	**0.8610**

The pre-2018 values are representative literature-reported benchmark values and may reflect different datasets, subsets, preprocessing procedures and evaluation protocols. The proposed model was evaluated on the paired test split of 1,143 image pairs.

The inference pipeline produces enhanced output images that reproduce the filename of each test input in the results/images directory, permitting direct visual comparison between the input and enhanced output images (Figure 3). A systematic qualitative review, for example a structured panel of representative input/output/reference triples selected to illustrate typical successes and failure modes, was not conducted as part of the documented project materials and is identified as future work rather than presented as a completed analysis.

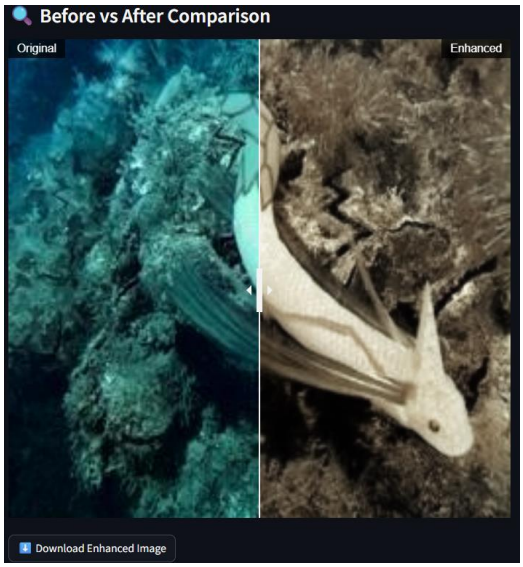

Figure 3. Representative qualitative comparison between the input underwater image and the enhanced output produced by the proposed U-Net model.

4.1 Application and deployment

The trained model is deployed through an interactive Streamlit web application (page title "Underwater Image Enhancement System") that provides a browser-based interface for applying the enhancement model to user-supplied images without requiring the user to invoke the training or inference scripts directly (Figure 4). The application accepts JPEG and PNG uploads and performs single-image inference; it does not process batches of images. A sidebar slider allows the working inference resolution to be adjusted between 128 and 512 pixels, with a default of 256 pixels, which differs from the fixed 256 × 256 resolution used in the standalone training and inference scripts, which expose no resolution control.

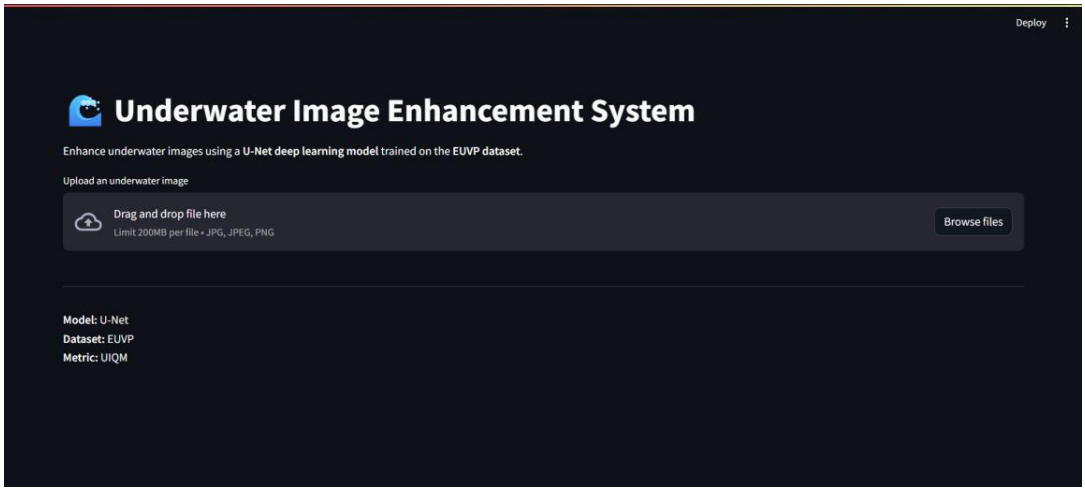

Figure 4. Streamlit-based application interface showing image-quality evaluation.

The application presents its output across three functional areas. A comparison view offers an interactive before/after slider alongside static side-by-side images and a button to download the enhanced result as a PNG file. A metrics view computes the custom UIQM-style score for both the original uploaded image and the enhanced output, displaying the original value, the enhanced value, their difference and a bar chart comparing the two. A details view reports the model name, the input resolution used and the processing device (CPU or GPU) selected for inference.

For a representative user-uploaded image processed through the application, the custom UIQM-style score was 210.84 for the original image and 226.83 for the enhanced output, an improvement of 15.99 (Figure 5). This example is an individual, application-level illustration only; it is produced by the same underlying metric family as the dataset-level test result reported in Table 4 (3.5692) but reflects a different image and a different aggregation (single image versus 1,143-image average), and it is not directly comparable in scale to that dataset-level figure.

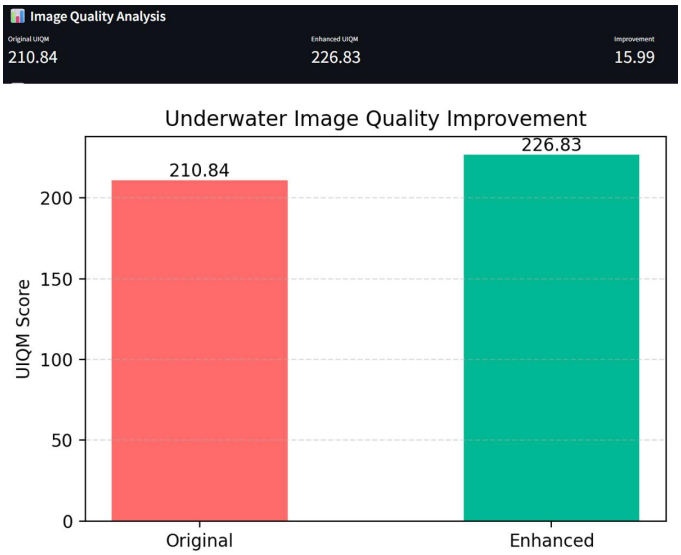

Figure 5. Example of the custom UIQM-style quality comparison displayed for a user-uploaded image.

Because a user-uploaded image has no associated reference image, the application does not compute PSNR or SSIM; only the no-reference UIQM-style metric is available within the interactive interface, applied separately to the original and enhanced images. This distinguishes the per-image, no-reference usage of the application from the standalone evaluation scripts, which compute PSNR and SSIM against reference images over the full test split and compute the UIQM-style score only on enhanced outputs, without an original-versus-enhanced comparison. A version of this application was previously made accessible through Streamlit Community Cloud; no additional hosting, containerization or cloud-infrastructure detail beyond the Streamlit-based application itself is claimed.

5. Discussion

The trained network achieved an average PSNR of 29.36 dB and an average SSIM of 0.8610 on the held-out test split under the adopted full-reference evaluation protocol, together with an average score of 3.5692 from the adapted no-reference UIQM-style measure.

As shown in Table 5, the proposed reduced-depth U-Net achieves favourable quantitative performance compared with representative pre-2018 underwater enhancement techniques and early deep-learning approaches. The comparison provides contextual evidence that the supervised encoder–decoder formulation can effectively learn the transformation from degraded underwater images to the corresponding target images. Unlike prior-based methods that rely on manually designed image-formation assumptions, the proposed approach performs end-to-end image-to-image regression using a reduced-depth U-Net architecture, trained with an L1 reconstruction loss and Adam optimization without adversarial training or a complex multi-component objective.

Several design choices merit discussion in light of the reviewed literature. The network is trained with an L1 objective only, with no adversarial term, distinguishing it from the pix2pix generator-plus-discriminator framework of Isola et al., to which it is otherwise architecturally analogous; this reflects a deliberate simplification rather than an attempt to match adversarially trained methods. [12] The reduced three-level depth, relative to the four-level depth of the original U-Net, may limit the receptive field and the capacity to model larger-scale spatial context. [9] The complete absence of data augmentation means that the exposure of the model to variation in pose, scale and colour balance during training is limited to whatever variation is naturally present in the 9,149 training pairs. Finally, the fixed five-epoch training schedule, adopted without a convergence check or early-stopping criterion, reflects a training-budget constraint rather than evidence that the model reached full convergence.

The current implementation is subject to further limitations identified directly from the documented training configuration and codebase. Training and standalone inference operate at a fixed 256 × 256 resolution, and only the Streamlit application exposes an adjustable resolution. The system depends entirely on paired input/reference training data and has not been evaluated in an unpaired or self-supervised setting, nor for generalization to EUVP subsets or underwater domains beyond those used here. No ablation study was conducted to isolate the contribution of individual architectural or training choices, and no controlled benchmark comparison against other underwater enhancement methods, traditional or learning-based, was performed. The custom UIQM-style implementation is a simplified adaptation of the canonical formulation and is not directly comparable to canonical UIQM values reported elsewhere. No downstream evaluation, such as object detection or segmentation performance on enhanced images, was performed. Finally, the training script does not fix a complete PyTorch/NumPy random seed, so exact training-time reproducibility beyond the dataset split is not guaranteed, and the training configuration itself was not the subject of systematic hyperparameter search.

Building on these limitations, several extensions are identified, none of which are implemented in the current system: longer and more systematic training guided by a convergence or early-stopping criterion; data augmentation strategies to increase effective training diversity; perceptual, SSIM-aware or other multi-component loss formulations alongside or instead of the current L1 objective; attention mechanisms or residual components within the encoder–decoder framework; transformer-based enhancement architectures; and multi-scale enhancement strategies with improved colour-restoration components.

Additional directions include support for higher-resolution processing beyond the current fixed or user-adjustable resolutions; extension to video-based underwater enhancement and real-time inference optimization; cross-dataset evaluation to assess generalization beyond the EUVP subset(s) used in this work; downstream evaluation on underwater object detection or segmentation tasks; and a systematic qualitative review of representative input/output/reference image panels.

Conclusion

This paper has documented the implementation, training configuration and evaluation of a supervised, reduced-depth U-Net-based framework for underwater image enhancement, trained on paired data from the EUVP dataset and deployed through an interactive Streamlit application. The system achieved an average PSNR of 29.36 dB, an average SSIM of 0.8610 and an average custom UIQM-style score of 3.5692 on a held-out test split of 1,143 image pairs, using an architecture and training procedure that intentionally avoid data augmentation, batch normalization, dropout, residual connections and adversarial training. The contribution of this work lies in the transparent implementation and evaluation of a reduced-depth U-Net

pipeline, together with an accessible interactive deployment. The quantitative results and contextual comparison indicate the potential of the proposed supervised approach, and the documented pipeline, spanning dataset organization, preprocessing, training, inference and evaluation, provides a clear foundation for reproducible experimentation. The limitations and future directions identified in the Discussion outline a concrete path toward more extensive baseline evaluation, cross-dataset validation and a more capable underwater image enhancement system.

References

1. Jaffe JS. Computer modeling and the design of optimal underwater imaging systems. IEEE J Ocean Eng. 1990;15(2):101-111. doi:10.1109/48.50695.
2. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600-612. doi:10.1109/TIP.2003.819861.
3. He K, Sun J, Tang X. Single image haze removal using dark channel prior. IEEE Trans Pattern Anal Mach Intell. 2011;33(12):2341-2353. doi:10.1109/TPAMI.2010.168.
4. Chiang JY, Chen YC. Underwater image enhancement by wavelength compensation and dehazing. IEEE Trans Image Process. 2012;21(4):1756-1769. doi:10.1109/TIP.2011.2179666.
5. Ancuti C, Ancuti CO, Haber T, Bekaert P. Enhancing underwater images and videos by fusion. In: Proc IEEE Conf Comput Vis Pattern Recognit (CVPR). 2012:81-88. doi:10.1109/CVPR.2012.6247661.
6. Drews P, Nascimento E, Moraes F, Botelho S, Campos M. Transmission estimation in underwater single images. In: Proc IEEE Int Conf Comput Vis Workshops (ICCVW). 2013:825-830. doi:10.1109/ICCVW.2013.113.
7. Drews PL, Nascimento ER, Botelho SS, Campos MFM. Underwater depth estimation and image restoration based on single images. IEEE Comput Graph Appl. 2016;36(2):24-35. doi:10.1109/MCG.2016.26.
8. Cai B, Xu X, Jia K, Qing C, Tao D. DehazeNet: an end-to-end system for single image haze removal. IEEE Trans Image Process. 2016;25(11):5187-5198. doi:10.1109/TIP.2016.2598681.
9. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015). LNCS vol. 9351. 2015:234-241. doi:10.1007/978-3-319-24574-4_28.
10. Yang M, Sowmya A. An underwater color image quality evaluation metric. IEEE Trans Image Process. 2015;24(12):6062-6071. doi:10.1109/TIP.2015.2491020.
11. Panetta K, Gao C, Agaian S. Human-visual-system-inspired underwater image quality measures. IEEE J Ocean Eng. 2016;41(3):541-551. doi:10.1109/JOE.2015.2469915.
12. Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In: Proc IEEE Conf Comput Vis Pattern Recognit (CVPR). 2017:1125-1134. doi:10.1109/CVPR.2017.632.