Underwater Image Color Correction Using Ensemble Colorization Network

Arpit Pipara
DA-IICT
Gandhinagar, Gujarat
201911030@daiict.ac.in

Urvi Oza
DA-IICT
Gandhinagar, Gujarat
201921009@daiict.ac.in

Srimanta Mandal
DA-IICT
Gandhinagar, Gujarat
in.srimanta.mandal@ieee.org

Abstract

Underwater image color correction has been gaining traction due to its usage in marine biology and surveillance. Color corrected images also help marine archaeologists in locating objects. The underwater image suffers from various degradation with respect to the depth at which the image is taken. In this paper, we propose an alternate path to correct the color of the underwater images. We address the problem of underwater image color correction as a colorization task. For this purpose, we propose a deep learning architecture that comprises of an ensemble encoder and a decoder. The ensemble encoder part uses pre-trained networks to extract multi-level features. These features are then fused together and are used up by the decoder to generate the color corrected output. We evaluate the performance of our model using reference-based as well as no reference-based metrics. The metrics indicate that the produced results are inline with the human perceptual system.

1. Introduction

Underwater image color correction has lately been the attraction due to its usage in marine biology and underwater surveillance. The images taken underwater suffer from color casts, noise, wavelength-dependent absorption, etc. Usually, the red color gets absorbed first [16] as it has a longer wavelength and lower frequency. The red color gets absorbed fully at around 5m depth [27]. This is followed by orange color, which vanishes at a depth of around 7-9m. Then yellow gets absorbed around at the depth of 16-18m, which is followed by green at a depth of around 28-30m. Thus, images usually appear to be greenish or bluish. Since the color disappears according to the depth, a color imbalance is reflected in the captured underwater image. Moreover, the captured image suffers from different degradations such as haze, noise, etc. due to scattering of light, non-uniform illumination etc.

The prevalent underwater image enhancement methods can be segregated into the following categories:

Supplementary Information Based Methods: The approach [23, 24, 25] relies on polarizing filters to improve the degraded image. Specialized radiation hardware is also employed to enhance the image.

Non-Physical Model-Based Methods: Iqbal *et al*. [14] work on stretching the pixel-values of non-dominant channels with respect to the dominant channel values. The intermediate output is then contrast corrected to get the final enhanced image. Ancuti *et al*. [3] proposed to fuse the contrast-enhanced image with a color-corrected image in a multi-scale fusion to get an improved result. Ancuti *et al*. [4] works in a two-step strategy where it combines white balancing technique and image fusion. This helps to compensate for the color caste while at the same time enhancing the details.

Physical Model-Based Methods: These methods work on the simple idea of "inverse". Li *et al*. [17] combines color correction and image dehazing to improve the visual quality of the degraded image. An algorithm is proposed to estimate the background light along with estimating medium transmission characteristics. Some researchers also employed optical properties of imaging in underwater environments. Carlevaris-Bianca *et al*. [5] exploits the difference in attenuation between color channels to estimate depth. It models the true scene as a Markov Random Field.

Data-Driven Models: With the onset of deep learning, the trend has shifted towards making end-to-end architectures for underwater image enhancement. For example, Li *et al*. [19] proposed WaterGAN, which works on generative adversarial networks along with a color restoration network. Li *et al*. [16] proposed WaterNet, which employs a gated fusion architecture. It works by combining the gamma-corrected image, histogram equalized image, and the white balanced image. The gamma correction and histogram equalization improves contrast and lighten up the darker regions. The white balance helps in correcting the color casts.

In general, the deep learning approaches try to modify the input raw underwater image according to the respective reference image during training. The objective is to minimize the error between the ground truth and the predicted output. At testing, no reference image is required.

However, we tackle the task of underwater image color correction differently. Cheng *et al*. [7] handles the colorization task using an ensemble of neural networks as a single neural network is not complex enough to tackle it alone. Cheng *et al*. [6] works by finding a very similar reference image for the image that needs to be colorized. We take inspiration from the model proposed by Li *et al*., WaterGAN[19], which modifies the color of an underwater image to an image that is taken in a natural setting. We also produce similar results but rather than modifying the RGB image, we handle this task as the conversion of grayscale image to colored one, i.e image colorization using DCNN. This approach yields results that contain natural colors. We start by converting RGB image to Lab colorspace and extract the L channel which is fed to the encoder as input. Ensemble encoder extract multi level features ranging from fine to coarse. These features are concatenated and then fed to the decoder which predicts the ab channel as the output.

The main contributions of the paper are:

- Exploring the underwater image color correction problem as a colorization task.
- Extraction of multi-level features using a pre-trained ensemble of CNNs.
- Evaluation of model with reference (PSNR & SSIM) and no-reference (UCIQE & UIQM) metrics.

The organization of our paper is as follows : Section 2 explains the optimal choice of color space, Section 3 provides the details of our architecture in detail. Experimentation and implementation details are laid out in section 4 along with the discussion of the results. The paper is then concluded in section 5.

2. Color Space

The most commonly used color spaces are RGB, YCbCr, Lab, etc. Every color space represents color differently. Hence, the choice of a particular task may affect the performance. For colorization task, RGB doesn't seem to be the appropriate color space as the model will have to predict all three channels. Whereas colorspaces like YCbCr and Lab which stores color information and luminance information separately requires the model to predict only two channels to colorize the grayscale image. Thus, decreasing the dimensional complexity. Among these two, Lab color space is optimal as it encompasses a wide range of colors. Most of the deep learning techniques [20, 26, 29, 30] in the image colorization domain have worked upon CIE Lab color space. The color space is segregated into the following components :

- L* (Lightness) component.
- a* (ranges from green to red) component.
- b* (ranges from blue to yellow) component.

The L* component ranges from 0 to 100, i.e., from black to white. The a* component has green on the negative axis and red on the positive axis. Similarly, the b* component has blue on the negative axis and yellow on the positive axis. Channels a and b ranges from -128 to +127. The center or the zero represents the grey color. Here the grayscale image acts as the L channel, which is fed to the network as input. The network is then required to predict the other two channels.

3. Proposed Approach

The proposed approach is unique in terms of tackling the problem. This is due to the reason that we are not using the Underwater Image Enhancement Benchmark (UIEB)[16] dataset for training. We have modeled this problem in an image colorization manner. The model is trained on the DIV2K[1, 2] dataset. The reason for doing this is once our model gets trained on the DIV2K dataset, the output produced will contain natural colors rather than bluish/greenish colors. So we train our model to predict natural colors in order to color the underwater images. This is the unique take on the problem.

In this section, we discuss the data pre-processing steps, the network design and the associated training parameters.

3.1. Data Pre-Processing

The DIV2K[1, 2] dataset comprises 800 high-definition training images. We start by extracting patches of size 224×224, as this is the desired size required by the pre-trained ensemble encoder. The reason for extracting patches is inclined towards the idea of minimizing the loss of information. If down-sampling would have been done, loss of information would have happened. These patches are then converted into Lab colorspace. After the transform, the L (luminance) component ranges from 0 to 100, while the 'a' and 'b' color components range from -128 to +127. In order to make these uniform, normalization is done. After normalization, the L (luminance) component ranges from 0 to 1, and the 'a' and 'b' color components range from -1 to +1. This normalization also boosts up the training and eradicates the exploding gradient problem. The L channel is then extracted and stacked three times before feeding to the ensemble encoder.

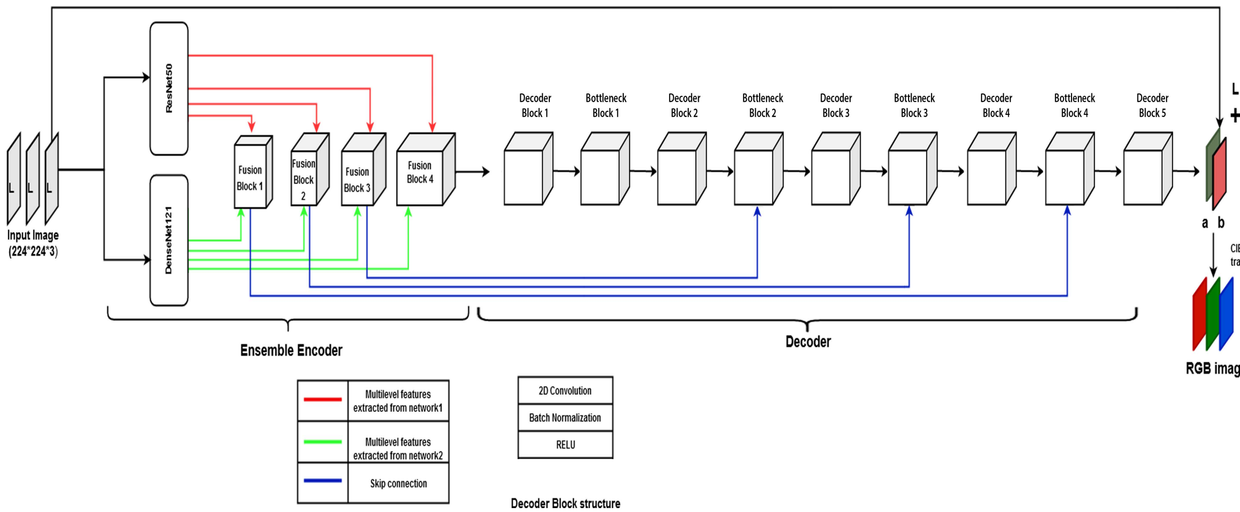


Figure 1. **Proposed ensemble encoder-decoder architecture.** Multi-level features are extracted at the ensemble encoder part. These features are then fused together. The fused features are used by the decoder to predict the ab channel.

3.2. Ensemble Encoder Design

Ensemble of pretrained ResNet50[12] and DenseNet121[13] networks without Fully Connected (FC) layers are used as encoder. Grayscale input image (size - $H \times W \times 3$) will be given to each of the networks separately. After analyzing features extracted at each layer of CNN, we divide the network into different stages based on intermediate layers features shape. Multiple features of the image will get extracted from each stage of single network. Following are the networks we used in encoder design:

- DenseNet121[13] - It is a densely-connected convolutional network which use dense blocks in its architecture to improve features propagation and to address the vanishing gradient problem. Based on the output size of dense blocks, we can divide DenseNet121 in 4 stages. We extract feature maps after dense block 1 of size 56×56, dense block 2 of size 28×28, dense block 3 of size 14×14 and dense block 4 of size 7×7.

- ResNet50[12] - It is a CNN architecture with skip connections between layers which helps in training very deep networks without facing the degradation problem. Resnet50 contains 5 stages each with a convolution and identity block which have different output sizes. We have extracted convolution layer output of different sizes - 56×56, 28×28, 14×14 and 7×7, to get fine to coarse feature maps.

After extracting level wise features, we have level 1 features of size 56×56, level 2 features of size 28×28, level 3 features of size 14×14 and level 4 features of size 7×7 from each network.

3.3. Fusion Strategy

Multi-level features coming from the ensemble encoder part need to be fused keeping in mind the dimensions associated with them. We have used concatenation as the feature fusion strategy. Before applying the fusion function, we perform a 1×1 convolution operation as a size transformation function on input feature maps to have a similar number of channels. So our fusion function can be generalized as $f^* : Y_1, Y_2, ..Y_n \rightarrow Y^*$, where $Y_1, Y_2, ...Y_n$ are feature maps from $Network_1, Network_2, ...Network_n$, respectively. As we have only two networks in ensemble, fusion function f^*generates output feature map

$$Y^* \in \mathbb{R}^{H^* \times W^* \times D^*}$$

by fusing

$$Y^1 \in \mathbb{R}^{H \times W \times D'} \text{and } Y^2 \in \mathbb{R}^{H \times W \times D'}$$

Here, the number of channels D' is the output of the number of channels of feature maps after applying the size transformation function. Concatenate fusion function concatenates the input feature maps at dimension D, so $D^* = D^1 + D^2 + ... + D^n$, and

$$Y^{concatenate} = Y^1 \| Y^2 \| ... \| Y^n$$

3.4. Decoder Design

The decoder design consists of a series of bottleneck and decoder blocks. The bottleneck block consists of multiple convolutional layers and batch normalization layers. ReLU has been used as the activation function. The bottleneck

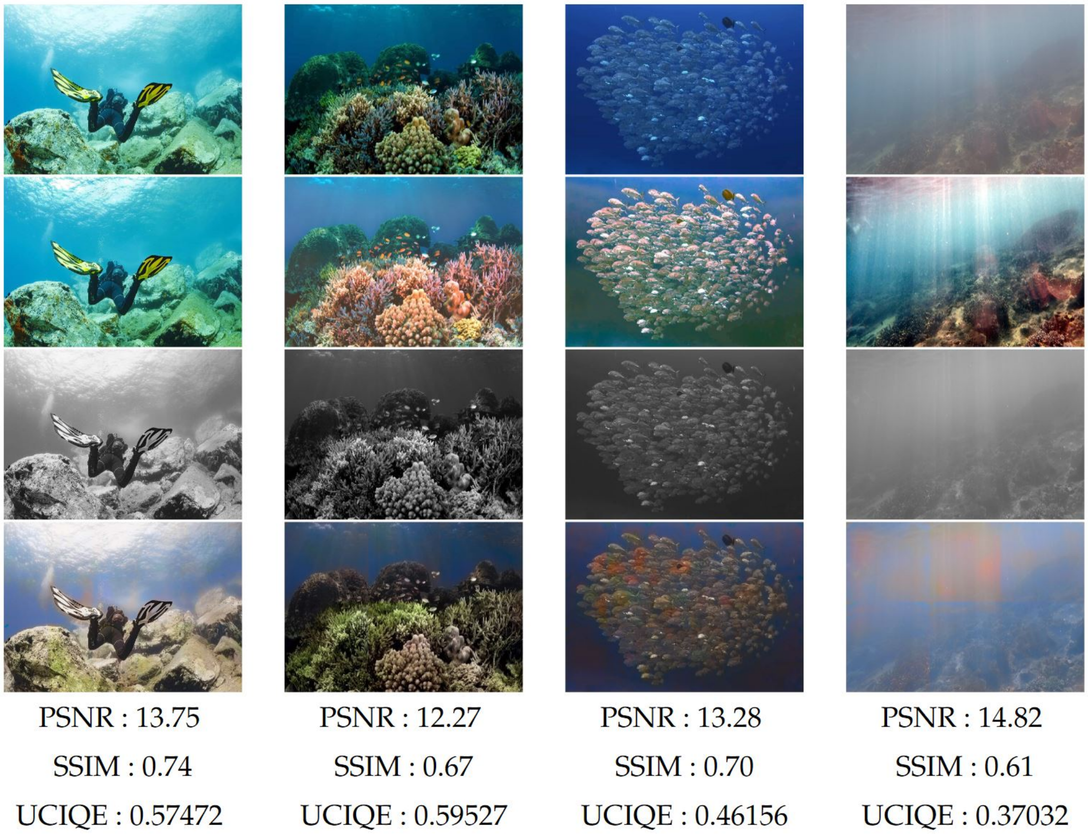


Figure 2. **Underwater Color Correction Visual Results with Non-Overlapping Patches.** The top row and the second row contains the raw underwater images and their respective color-corrected reference images. The third row contains the input that is fed to the model. The last row contains the output generated by our model along with the respective scores.

blocks help to get the representation of the input in reduced dimensionality. The decoder blocks are made up of convolution layer, batch norm layer and upsampling layer. The last decoder block is different from the rest as it uses TanH as the activation function. As TanH is used at last, the output produced is in the range of [-1,1]. All the blocks receive input from the previous blocks. In addition, the bottleneck blocks receive respective fused features from the fusion block as the skip connection. These skip connections help to strengthen the information flow and also mitigates the issue of vanishing gradient. Finally, the predicted ab channel is integrated with the L channel to get the desired output.

3.5. Loss Function and Training

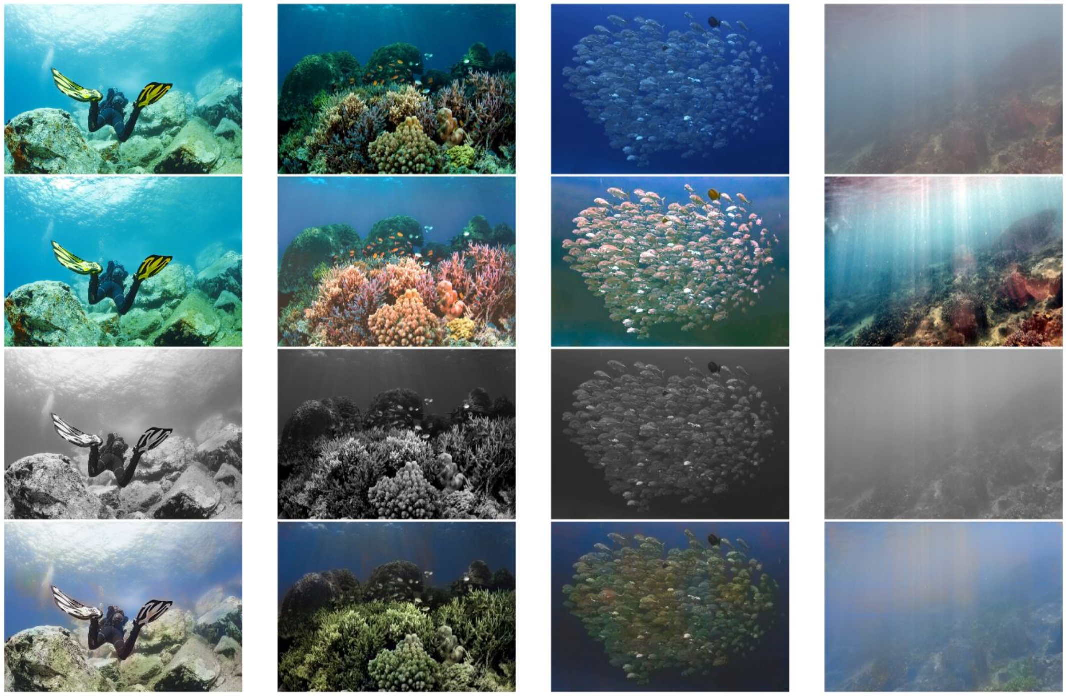
The encoder module uses pre-trained networks. Their weights are freezed during the training so no weight updation takes place. Now, as only the decoder part needs to be trained, it significantly expedites the training process. The predicted ab channels are compared with the ground truth ab channels using Mean Square Error(MSE) as the loss function. Adam optimizer is used to handle the optimization part. The learning rate is set to 0.001, along with the batch size of 8. The network is trained for 45 epochs where 1 epoch took around 3 hours to finish.

4. Experiment

4.1. Dataset

Deep learning requires a large-scale database to make the model work, but the generation of color corrected images from underwater images is a manual and time-consuming task. The dataset Underwater Image Enhancement Benchmark Dataset (UIEB)[16] has been generated by showing the users a variety of color-corrected images. The dataset comprises of 950 real underwater images, out of which 890 have reference images, and the rest 60 images are considered the test images.

However, as we have modeled the problem differently, we have used the DIV2K[1, 2] dataset as the training dataset. It has 800 high-definition RGB images for training and 100 images for validation. For testing our model, 890 training images and 60 testing images from the Underwater Image Enhancement Benchmark Dataset and Beyond



PSNR : 13.75 SSIM : 0.78 UCIQE : 0.57472

PSNR : 18.99 SSIM : 0.74 UCIQE : 0.60413

PSNR : 15.90 SSIM : 0.83 UCIQE : 0.47394

PSNR : 14.37 SSIM : 0.60 UCIQE : 0.40474

Figure 3. **Underwater Color Correction Visual Results with Overlapping Patches.** The top row and the second row contains the raw underwater images and their respective color-corrected reference images. The third row contains the input that is fed to the model. The last row contains the output generated by our model along with the respective scores.

are used.

4.2. Implementation Details

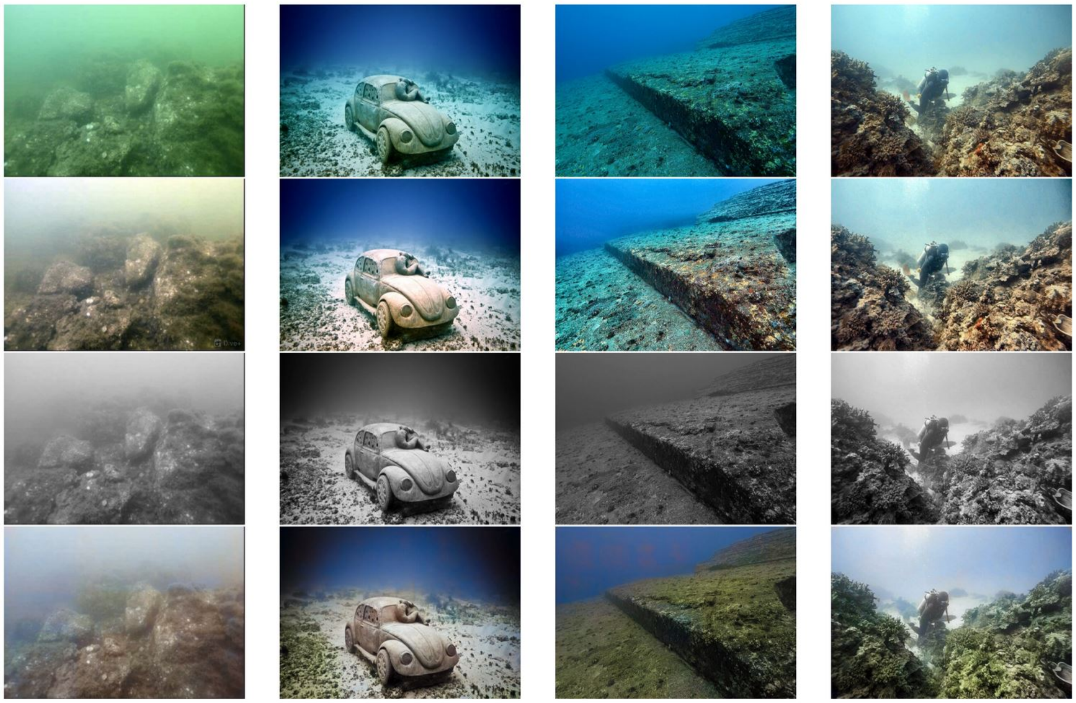
The network, as stated above, is trained upon the DIV2K dataset. We start by extracting patches of size 224×224 with a step size of 1.0. The step size of 1.0 indicates that there are no overlapping patches. These patches are then fed to the ensemble encoder part, and the decoder part predicts the output. At the time of testing, we extract patches of the same size but with an overlap of 25%. This is done as the UIEB dataset has majority of images which contains tiny objects, and, hence artifacts are observed in the results. On the contrary, the DIV2K dataset does not contain a lot of images with tiny objects. The overlap portion is averaged out at the end.

4.3. Results

We started out by testing the model on the UIEB training dataset. The grayscale patches of 224×224 are given as input to the model. For the initial testing of the model, non-overlapping patches are used. The results are shown in Fig. 2. As expected, the underwater greenish/bluish color has been replaced with natural colors. The final result suffers from artifacts when there are too many small objects present. In this case, the model is not able to give out a clear distinction between those objects.

To overcome the artifacts issue, we use overlapping patches with an overlap of 25%. The overlapped portion is then averaged out at the end during the reconstruction of the full image from the patches. The results generated by overlapping patches are shown in Fig. 3. It is seen that the artifacts that were present have now been significantly reduced due to overlapping patches. The averaging operation on the overlap part helps to reduce the colors that are seen on the leftmost and rightmost part of the axis. The bottom right image in Fig. 2 shows color patches which are seen on the leftmost and rightmost part of the axis of 'a' and 'b' channel, while with overlapping patches these kind of colors are averaged out as seen in Fig. 3. The model is able to give out results as if the images are taken in a natural setting. These results can be helpful during underwater surveillance as they will help to identify objects quickly. More visual results are shown in Fig. 4.

In the Fig. 5, all the underwater images have almost the

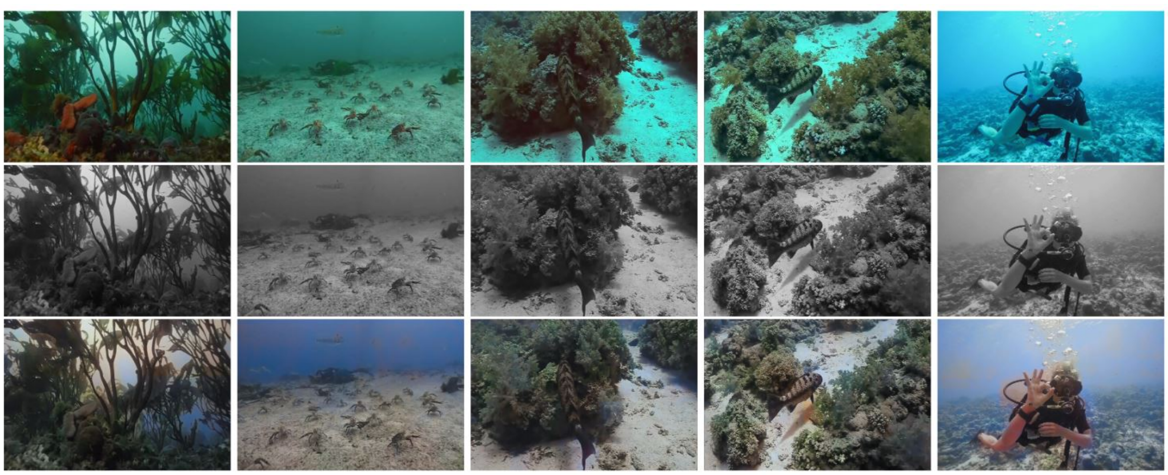


PSNR : 21.53 PSNR : 18.12 PSNR : 11.69 PSNR : 21.10

SSIM : 0.92 SSIM : 0.87 SSIM : 0.54 SSIM : 0.90

UCIQE : 0.50979 UCIQE : 0.59913 UCIQE : 0.55406 UCIQE : 0.58629

Figure 4. **Underwater Color Correction Visual Results with Overlapping Patches.** The top row and the second row contains the raw underwater images and their respective color-corrected reference images. The third row contains the input that is fed to the model. The last row contains the output generated by our model along with the respective scores.



UCIQE : 0.55482 UCIQE : 0.49688 UCIQE : 0.49394 UCIQE : 0.52363 UCIQE : 0.54634

Figure 5. **Underwater Color Correction Visual Results.** The top row contains the underwater raw images. The middle row contains the the images given to the model as input. The bottom row contains the outputs produced by the model. These results are obtained on the UIEB[16] test dataset. The input images have the same color tone for the background and the objects present, while the output produced shows better differentiation.

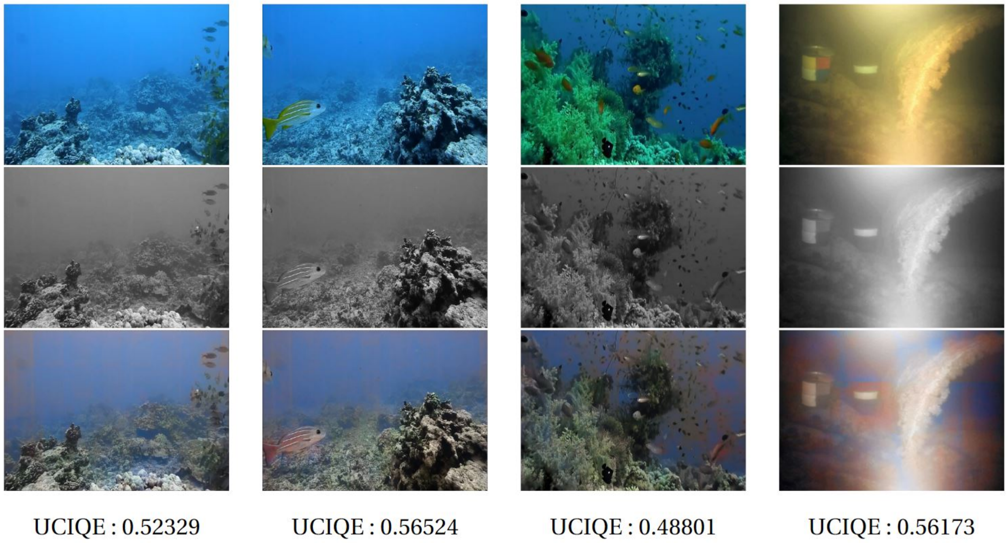


Figure 6. **Underwater Color Correction Visual Results.** The top row contains the underwater raw images. The middle row contains the the images given to the model as input. The bottom row contains the outputs produced by the model.

same color tone. There is little to no difference between the color tone of different objects present in the image. Whereas the outputs produced by our model show better differentiation among the objects. Hence, the task of locating objects becomes convenient.

The last column of Fig.6 contains non-uniform illuminated underwater image. The Fig.7 shows visual results of our approach along with other approaches like ARC[11], DBL[18], UWCNN[15] and WaterNet[16]. ARC tries to recover color for shorter wavelengths. In column 2, it can be seen that all the different approaches, except WaterNet[16], produce results that have little or no change compare to the raw underwater image. The color tone of the whole image is the same. The same is seen for other results as well. This is not the case with our result. The fish and its surroundings are colored differently showing a visible differentiation. In the last column, UWCNN[15] has produced an abrupt yellow-green color which is nowhere to be seen in our results. This in turn makes our results more suitable for object localization.

The metrics used to evaluate the results broadly fall into two categories: Reference-based and No reference-based

Reference-based: PSNR and SSIM are the standard reference-based metrics. The table 1 shows the PSNR and SSIM values obtained on the UIEB training set. The reason for low PSNR and SSIM is that our model has been trained on the DIV2K dataset which does not contain any degradation, so the model does not do anything specific to handle the degradations present. Hence, a low PSNR and SSIM is the expected output.

No Reference-based: UCIQE (Underwater Color Image Quality Evaluation metrics) [28] and UIQM [21] are used to evaluate the quality of underwater images. A high value of UCIQE indicates the result is highly color balanced in terms of saturation, chroma, and contrast. Similarly, a high value of UIQM suggests that the results are more in line with the human visual system. The table 2 shows the UIQM and UCIQE values obtained on the UIEB training dataset. It is seen that our model is able to perform better than all the listed approaches with respect to UIQM metric. This indicates that the results are significantly are in the line with the human perceptual system.

Metric	**Value**
PSNR	16.31
SSIM	0.76

Table 1. Reference-based metrics results on the UIEB training set used as testing set.

Methods	**UCIQE**	**UIQM**
Fusion-based [3]	0.6414	1.5310
Two-step based [9]	0.5776	1.4002
Retinex-based [10]	0.6062	1.4338
UDCP [8]	0.5852	1.6297
Regression-based [17]	0.5971	1.2996
GDCP [22]	0.5993	1.4301
Ours	0.5013	3.3363

Table 2. No reference-based metrics results on the UIEB training set used as testing set.

The average UCIQE and UIQM scores obtained on the UIEB testing set are 0.4735 and 1.9006, respectively. The visual results on the UIEB test dataset are shown in Fig. 5 and 6. The results depict that although the images have been color corrected in a way that they are captured in a natural

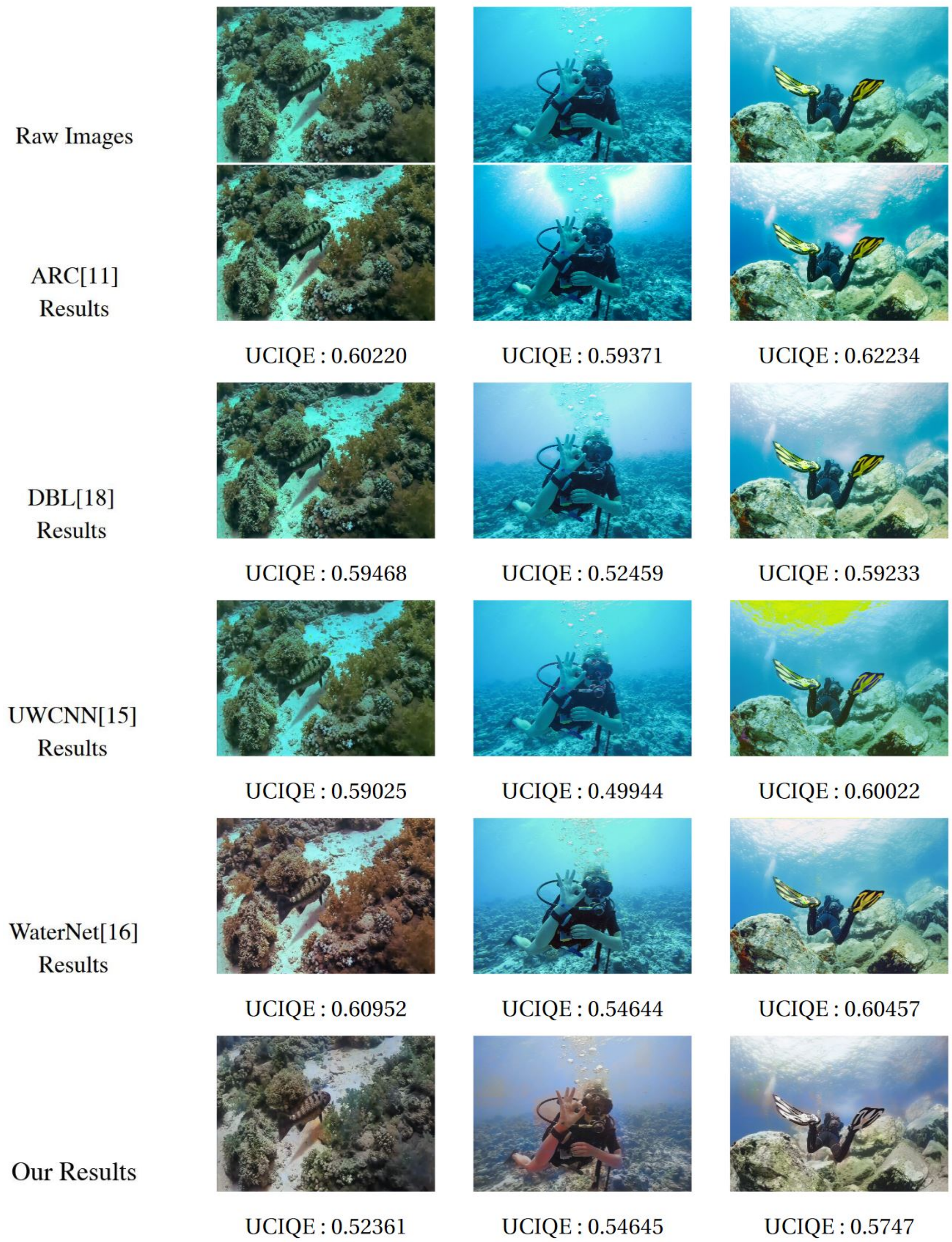


Figure 7. **Visual Results Comparison with Different Approaches on the UIEB Dataset**. The top row contains the raw underwater images. Various other rows contains results of the respective approaches and the last row contains the results obtained by our approach.

surrounding but the degradations like blurring, haze, etc., are still there.

5. Conclusion

In this paper, a simple ensemble encoder-decoder based colorization network has been laid out. The colorization architecture is used to handle the underwater color correction problem. The obtained results show better differentiation between the background and the different objects present in the image. The UIQM score justifies that the results obtained are inline with the human visual system. This can be useful for underwater archaeologists as the results will decrease the complexity of locating an object.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[3] Cosmin Ancuti, Codruta Orniana Ancuti, Tom Haber, and Philippe Bekaert. Enhancing underwater images and videos by fusion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 81–88. IEEE, 2012.

[4] Codruta O Ancuti, Cosmin Ancuti, Christophe De Vleeschouwer, and Philippe Bekaert. Color balance and fusion for underwater image enhancement. *IEEE Transactions on image processing*, 27(1):379–393, 2017.

[5] Nicholas Carlevaris-Bianco, Anush Mohan, and Ryan M Eustice. Initial results in underwater single image dehazing. In *Oceans 2010 Mts/IEEE Seattle*, pages 1–8. IEEE, 2010.

[6] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[7] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Colorization using neural network ensemble. *IEEE Transactions on Image Processing*, 26(11):5491–5505, 2017.

[8] Paulo LJ Drews, Erickson R Nascimento, Silvia SC Botelho, and Mario Fernando Montenegro Campos. Underwater depth estimation and image restoration based on single images. *IEEE computer graphics and applications*, 36(2):24–35, 2016.

[9] Xueyang Fu, Zhiwen Fan, Mei Ling, Yue Huang, and Xinghao Ding. Two-step approach for single underwater image enhancement. In *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 789–794. IEEE, 2017.

[10] Xueyang Fu, Peixian Zhuang, Yue Huang, Yinghao Liao, Xiao-Ping Zhang, and Xinghao Ding. A retinex-based enhancing approach for single underwater image. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4572–4576. IEEE, 2014.

[11] Adrian Galdran, David Pardo, Artzai Picón, and Aitor Alvarez-Gila. Automatic red-channel underwater image restoration. *Journal of Visual Communication and Image Representation*, 26:132–145, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[14] Kashif Iqbal, Michael Odetayo, Anne James, Rosalina Abdul Salam, and Abdullah Zawawi Hj Talib. Enhancing the low quality images using unsupervised colour correction method. In *2010 IEEE International Conference on Systems, Man and Cybernetics*, pages 1703–1709. IEEE, 2010.

[15] Chongyi Li, Saeed Anwar, and Fatih Porikli. Underwater scene prior inspired deep underwater image and video enhancement. *Pattern Recognition*, 98:107038, 2020.

[16] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing*, 29:4376–4389, 2019.

[17] Chongyi Li, Jichang Guo, Chunle Guo, Runmin Cong, and Jiachang Gong. A hybrid method for underwater image correction. *Pattern Recognition Letters*, 94:62–67, 2017.

[18] Chau Yi Li and Andrea Cavallaro. Background light estimation for depth-dependent underwater image restoration. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1528–1532. IEEE, 2018.

[19] Jie Li, Katherine A Skinner, Ryan M Eustice, and Matthew Johnson-Roberson. Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images. *IEEE Robotics and Automation letters*, 3(1):387–394, 2017.

[20] Gokhan Ozbulak. Image colorization by capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[21] Karen Panetta, Chen Gao, and Sos Agaian. Human-visual-system-inspired underwater image quality measures. *IEEE Journal of Oceanic Engineering*, 41(3):541–551, 2015.

[22] Yan-Tsung Peng, Keming Cao, and Pamela C Cosman. Generalization of the dark channel prior for single image restoration. *IEEE Transactions on Image Processing*, 27(6):2856–2868, 2018.

[23] Yoav Y Schechner and Nir Karpel. Clear underwater vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.

[24] Yoav Y Schechner and Nir Karpel. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of oceanic engineering*, 30(3):570–587, 2005.

[25] Nadav Shashar, Roger T Hanlon, and Anne deM Petz. Polarization vision helps detect transparent prey. *Nature*, 393(6682):222–223, 1998.

[26] Jheng-Wei Su, Hung-Kuo Chu, and Jia-Bin Huang. Instance-aware image colorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7968–7977, 2020.

[27] Yan Wang, Wei Song, Giancarlo Fortino, Li-Zhe Qi, Wenqiang Zhang, and Antonio Liotta. An experimental-based review of image enhancement and image restoration methods for underwater imaging. *IEEE Access*, 7:140233–140251, 2019.

[28] Miao Yang and Arcot Sowmya. An underwater color image quality evaluation metric. *IEEE Transactions on Image Processing*, 24(12):6062–6071, 2015.

[29] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[30] Hengyuan Zhao, Wenhao Wu, Yihao Liu, and Dongliang He. Color2style: Real-time exemplar-based image colorization with self-reference learning and deep feature modulation. *arXiv preprint arXiv:2106.08017*, 2021.