Exploring Underwater Image Enhancement

Ryan Zhang
Princeton University
ryanz@princeton.edu

Howard Yen
Princeton University
hyen@princeton.edu

Abstract

Underwater image enhancement seeks to augment the image quality of photos taken underwater by correcting for artifacts like backscatter, light attenuation, and other sources of haze. While traditional approaches have employed complex equations to model and correct these phenomena, a simple neural approach that is flexible and can generalize to many settings is also desirable. To address this task, we explore both a two-stage convolutional network based approach, as well as a large-scale pre-trained transformer based approach. The two-stage approach is split into grayscale enhancement and colorization networks, while the transformer network (DALL-E) is modified for our purposes. We demonstrate results on common underwater image datasets, and discuss experiments, interesting results, and sources of error.

1. Introduction

Taking images in an underwater setting introduces artifacts and noise, which can lead to blurry images with inaccurate colors. There are a number of factors at play leading to this. First, due to particles floating in deep water, images are subject to the effects of backscatter, which may appear as noise. Furthermore, light is able to travel deeper in water depending on its wavelength, thus different depths will be dominated by different colors while some objects are left virtually color-less. As such, correcting these images automatically presents a unique challenge.

While this task has historically been approached by composing mathematical models for scattering and absorption, [2] it can also be approached with modern deep learning. It is, of course, related to non-underwater image enhancement, but also can draw inspiration from tasks like image colorization and image denoising, where the output image is closely related to the input image. Unfortunately, the lack of high-quality ground truth pairs poses a major challenge for deep learning models. Unlike in-air images, it is much more difficult to collect realistic and representative underwater images along with their corrected versions. Previous works attempt to remedy this by artificially adding noise and color to input images, some via generative models like UW-GAN [23]. However, it has been shown that models trained on these generated images do not always translate well to the real-world and are limited in their generalization ability [2].

In this project, we investigate a two-step approach to this task. The proposed method consists of a gray-scale enhancement network to first sharpen and lighten the image, followed by a colorization network to reapply colors. Both components are based on Convolutional Neural Networks (CNNs), and the output of the first component is used as input to the second. By splitting the model into two parts, we aim to simplify the problem into simpler subtasks in hopes of addressing the issue of data scarcity. We conduct experiments on the UIEB dataset [15] to make comparisons standard across the literature, and compare results with other neural models to demonstrate the strengths and weaknesses of our approach. We train our model on the EUVP dataset [11], which consists of approximately 11,000 image to ground truth pairs.

Another approach that we pursue is using pre-trained autoregressive transformer models. Due to their impressive results on image generation tasks, we are curious to see their results when applied to this task. We aim to use a encoder-decoder architecture that takes advantage of transformer layers to achieve robust interaction between all parts of the image. Similar to our other proposed approach, we will also first train on the EUVP dataset and evaluate on the UIED dataset.

2. Related Works

2.1. Image-to-Image Networks

There is a huge variety of image-to-image translation tasks that employ deep learning, ranging from image inpainting to style transfer. As such, there is a wealth of model schemes to choose from. While some models optimize on a distance metric–like Mean-Squared Error or L1 Loss–to the target image, others, like [25] treat image colorization as a classification problem. Model architectures, like the

classic U-Net [19] or a more general encoder-decoder structure play an important role as well. Beyond this, generative models like Variational Autoencoders and Generative Adversarial Networks (GAN) have played important roles in tasks like image super-resolution and style transfer, among others [14]. More recently, transformer [13] and diffusion models [8] have shown great promise in this area. In this project, we explore a number of image-to-image translation approaches.

2.2. Underwater Image Enhancement

Historically, techniques have made use of equations to model and correct haze caused by scatter in underwater images. Recently, Akkaynak et al. detailed a complex underwater image formation model that takes into account variables like light absorption in water, scattering, and light attenuation. Such a model requires detailed and precise calculation; thus it is also desirable to search for neural models that can effectively encode these variables.

Neural network based models for underwater image enhancement generally take a CNN or GAN approach [2], with both producing realistic results under certain settings [4, 10, 9]. Models largely differ in their architectures, with many incorporating additional information, such as depth map estimations, or auxiliary tasks to better optimize for diverse settings. However, as mentioned previously, a pressing issue for each of these approaches is the scarcity of good training data.

2.3. Transformers

In the recent years, transformer-based large pre-trained models like BERT [7] and GPT3 [3] have found success in many natural language processing tasks by training on extremely large corpora [22, 18]. The main advantages of these models are that they are able to efficiently train in parallel on corpora that contains billions of text tokens. Due to their successes in natural language, similar architectures and training set ups have been applied to computer vision tasks like image captioning and image generation [5]. For instance, the integration of text and visual features in BERT-like models such as ViLBERT, VisualBERT, and VLBERT [17, 16, 20] have shown great performances on tasks like Visual Question Answering [1] and Visual Commonsense Reasoning [24]. Similarly, autoregressive transformer models like ImageGPT [6] have shown success in generating realistic pictures given some text and image features as inputs.

For this project, we want to apply DALL-E [18] to this task. DALL-E is a variant of GPT-3 [3] that was trained to generate images from text descriptions, and it showed tremendous capabilities to generate plausible images based on just text prompts as well as abilities to apply transformations on existing images. The model first takes the input image and compress it into a 32x32 grid of discrete tokens by using a Discrete Variational Autoencoder (dVAE), which effectively parses the input image into 1024 discrete tokens. Then, it concatenates the image tokens with input text tokens and feeds all the tokens into an autoregressive transformer, which allows the model to learn a strong representation of both visual and text tokens and their joint distribution over the visual outputs. DALL-E's performance on image generation exceeded that of other popular image generation models like DF-GAN[21], and given its vast pre-training corpus, may have already learned information helpful to transfer to this task.

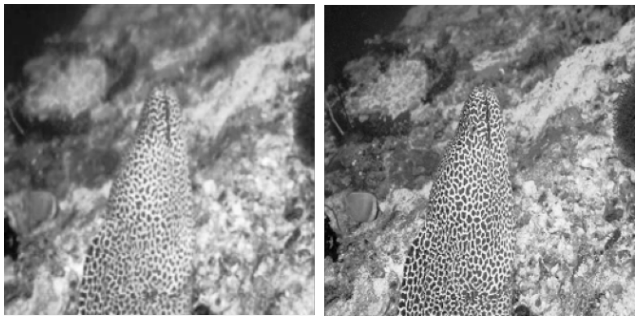

Figure 1. Original un-enhanced image converted to grayscale, left, and ground truth image converted to grayscale, right. We observe that edges in the right image appear more clearly than those in the left image due to blurring

.

Loss Function	MSE	SSIM	PSNR
Average	174.42	0.9052	27.211

Table 1. Difference between original and ground-truth grayscale images

3. Model Design

3.1. Two-stage Design

3.1.1 Grayscale Sharpening

Manually examining the grayscale images for the original image and the ground truth suggests that a simple first task is to enhance them to offset some of the distortion and blurriness present, as seen in Figure 1. In Table 1, we observe that there is a quantitative difference between the two images, indicating that there are gains to be made in this channel. Since this simplifies the task to working only with a single channel, we hope that a model can specialize in translating between original and output more easily and with less data. We experiment with a number of architectures and loss functions for this model, the results of which are detailed in Section 4.

3.1.2 Image re-colorization

Given an enhanced grayscale image, we need to reapply colors appropriately. Unlike traditional image colorization tasks, though, we are given additional information by the original image. More specifically, some of the colors in the original image with remain in the final image, while others may be altered but still be dependent on the original input. For those objects that have muted colors due to light attenuation, we face the issue of color ambiguity. This problem is also common in other image colorization tasks, and occurs when objects in the image may plausibly be colored in multiple ways. This means that traditional loss functions, like MSE loss, harshly penalize painting a car red versus blue, even though both are plausible and not necessarily incorrect without extra information.

To address these challenges, we use Zhang et al.'s implementation of Colorful Image Colorization [25] [1]. Rather than optimizing for MSE Loss, the authors treat colorization as a classification problem, where the task is to predict one of 313 color bins rather than pixel value. They further balance values by class to more heavily favor rare colors, which is desirable in our case as many of the image pixels will be dominated by water. Since we need to condition the model on the input colors, we modify the network to accept the original image rather than only the gray channel.

Overall, this model setup lends itself well to the task. Image colorization primarily happens in the LAB color space, where the L channel represents the lightness and is the grayscale image, and the AB channels encode image color. Thus, we can easily apply the corrected grayscale image to the colorization network by appending it to the AB channels. The colorization network is only concerned with assigning colors, rather than correcting how the image looks, leading to a further simplified task. As we see in Section 4, we observe results that are competitive with previous works.

3.2. DALL-E

In order to take advantage of the already pre-trained weights for the DALL-E model, we leave the model architecture mostly the same. However, we remove the text component of the model, as we are only processing images for this task. The model is consisted of two main componenets: the encoder and the decoder. The encoder takes an image of size 256 pixels by 256 pixels and compress it into a 32x32 grid where each token maps to 8192 possible values. Then, the decoder takes the image tokens and uses an autoregressive transformer to generate the output image. In total, the model contains around 10 million parameters.

For training, we first load OpenAi's pretrained weights[2]. Then, we train on the Paired section of the EUVP dataset across all three categories: underwater dark, underwater imagenet, and underwater scenes, where we scale every image to the size of 256 pixels by 256 pixels. We use a learning rate of 5×10^{-5} due to the large size of the model and a weight decay of 1×10^{-5} for regularization. Since the model is already pre-trained, we fine-tuned it on the task by training it for just 1 epochs across the entire dataset. For the loss function, we tested the results of three different loss functions: mean squared error (MSE) loss, L1 loss, and multi-scale structural similarity, which introduces an additional calibration for the parameters defining the importance of different scales (MSSSIM) loss[3]. We decided to try out three different types of loss because when training an image generative model, it is not always immediately obvious which loss function will perform the best.

4. Results

Results are quantitatively compared using Mean Squared Error, Peak signal-to-noise ratio (PSNR), and Structural Similarity Index (SSIM). While MSE loss are concerned with pixel-to-pixel differences between the output and ground truth, PSNR and SSIM are different ways to measure the perceived quality of an image given a reference, and are commonly used in other underwater image colorization approaches. In general, lower values are better for MSE, while higher values are better for PSNR and SSIM, with SSIM reaching a maximum value of 1 and PSNR reaching a maximum of 100.

4.1. Two-Stage Network

4.1.1 Grayscale Enhancement

The grayscale enhancement network is built in a fairly straightforward manner, with convolution blocks separated by batch normalization and dropout layers for additional regularization. All models are trained until plateau on a hold-out validation set. Images are converted to grayscale and normalized to between 0 and 1.

We first test by optimizing on $\mathcal{L}_{MSE}$, $\mathcal{L}_{L1}$, $\mathcal{L}_{MSSSIM}$, and $\alpha * \mathcal{L}_{MSSSIM} + \beta * \mathcal{L}_{MSE}$. We repeat the MSE experiment by sharpening the image prior to model training to enhance edges and simplify the task for the model. Finally, we experiment with a Pix2Pix GAN model [4] to test if an adversarial environment is suited for the task.

We first detail some of the more promising qualitative examples that came up during our experimentation in Fig-

[1] https://github.com/Time0o/colorful-colorization

[2] https://github.com/openai/dall-e

[3] https://github.com/jorge-pessoa/pytorch-msssim

[4] https://github.com/eriklindernoren/PyTorch-GAN

Loss Function	MSE ($\times 10^3$) $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
MSE	1.0609	0.8216	19.815
MSSSIM	1.736	0.7838	17.667
MSE + MSSSIM	1.3193	0.8349	18.769
Sharpen + MSE	1.009	0.8294	20.358

Table 2. Quantitative comparison of test set metrics after training with different loss terms on grayscale image correction.

ure 2. We observe that the MSSSIM optimized model tends to leave bright-spots in the image, making some areas undetailed as regions are washed out by the white color. The model optimized on MSE and with sharpening appears the most ideal, while the GAN model tends to leave artifacts in the image in the form of white splatters. Overall, models are able to preserve image characteristics and enhance the original to some degree. We next compare the highest scoring models quantitatively in Table 2.

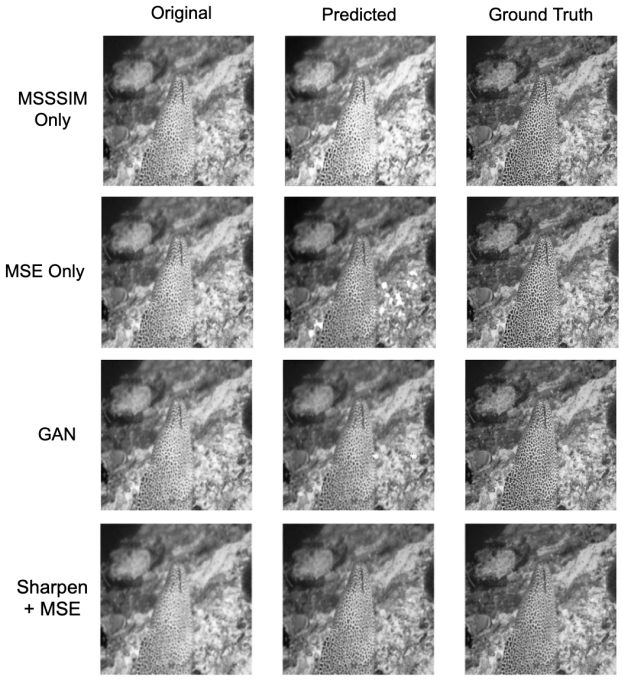

Figure 2. Qualitative results for grayscale enhanced images on held out image

4.1.2 Colorization Network

As mentioned previously, we adopt Zhang et al.'s [25] colorization network and modify it to handle 3 channel inputs. We first train model M2, and give the model perfectly reconstructed grayscale images by using the gray channel of the ground truth image as input, while maintaining the AB channels of the unenhanced image as the other two channels. The model outputs the AB channels for the enhanced image. At test time, the output of the grayscale enhancement network is used.

We observe in Figure 3 that the colorization output is sensitive to errors in the grayscale input, generating completely different colors when the grayscale image contains noise. As such, we train model M1, which during training randomly selects either the ground-truth grayscale image, or the un-enhanced grayscale input. This is meant to make the model more robust to cases where the input does not perfectly match with the output. We report quantitative results on each of these experiments in Table 3. From the results, we see that the weak link in this system is the grayscale enhancement network, and gains in grayscale image quality would lead to much better results.

Model	MSE ($\times 10^3$) $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
M1, Perfect	0.6201	0.9026	21.465
M2, Perfect	0.6090	0.9043	21.597
M1, Imperfect	1.3635	0.7715	17.683
M2, Imperfect	1.3365	0.7719	17.7147
DUIENet [15]	1.0122	0.8093	19.29
DenseGAN [10]	1.3636	0.4239	17.56

Table 3. Quantitative results on the test set for two-stage underwater image enhancement. Any row with "Perfect" is best-case scenario and nearly impossible to achieve in the real world. We also show some results for other approaches.

Loss Function	MSE ($\times 10^3$) $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
L1	1.6353	0.6000	16.2434
MSE	1.6463	0.5992	16.2139
MSSSIM	9.1409	0.6518	19.1060

Table 4. Results of DALL-E models on the EUVP (training) dataset [12], separated by the loss function that it was trained on.

Loss Function	MSE ($\times 10^3$) $\downarrow$	SSIM $\uparrow$	PSNR $\uparrow$
L1	1.6666	0.5840	16.6303
MSE	1.6734	0.5830	16.6107
MSSSIM	1.3352	0.6085	17.6112

Table 5. Results of DALL-E models on the UIED (evaluation) dataset [15], separated by the loss function that it was trained on.

4.2. DALL-E

We evaluate the DALL-E models trained on different losses on the UIED dataset. As shown in Table 4 and Table 5, DALL-E trained using MSSSIM as the loss function performs the best across all three metrics on both the training set and the evaluation set. Even though we would typically expect the model trained using MSE as the objective

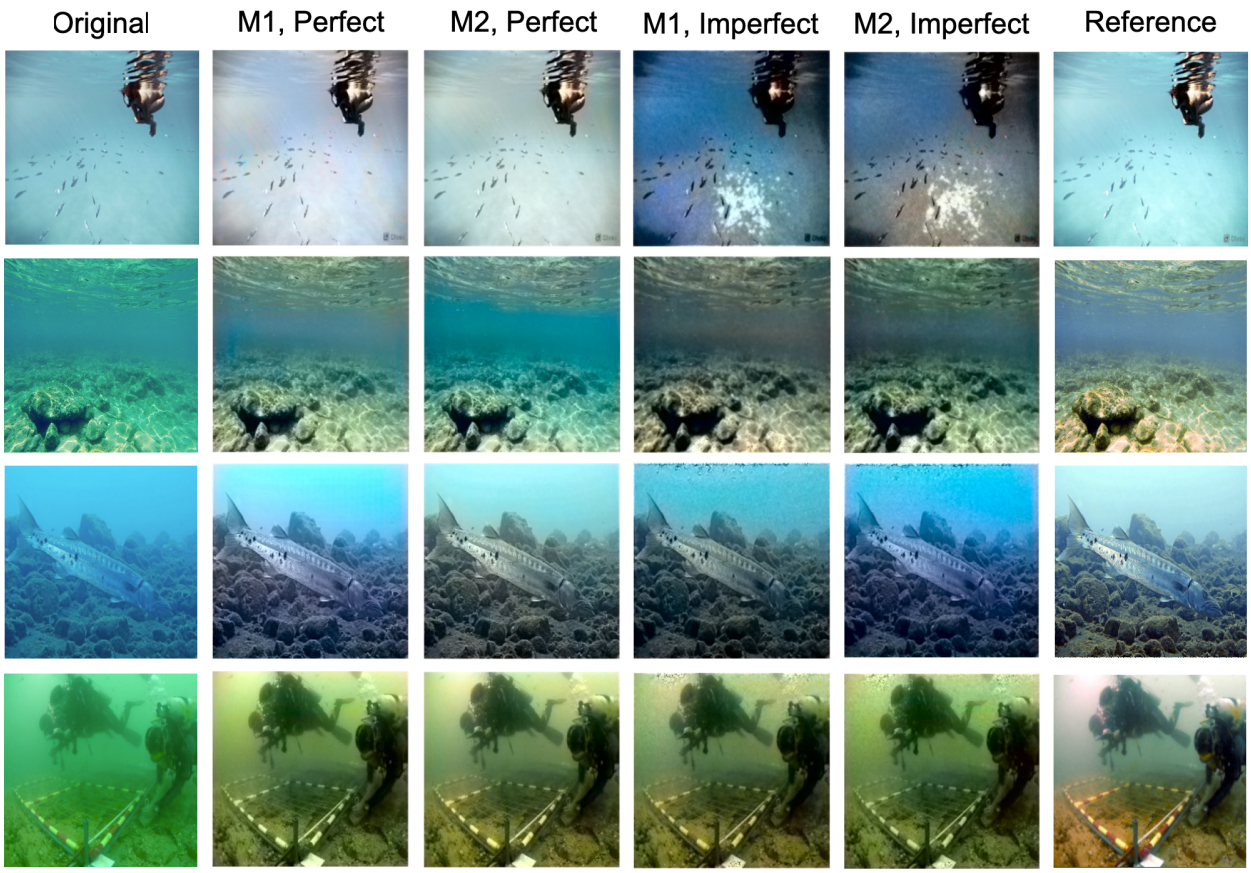

Figure 3. Qualitative results on the test set for full-model images with the two stream network. M1 is trained with a random set of ground truth and noisy grayscale images, and M2 is trained with ground truth grayscale images. Perfect indicates that the model is given the ground truth grayscale at test time, while imperfect indicates that it was given the output from the grayscale enhancement network.

function would have the lowest MSE, the DALL-E model trained using MSSSIM actually achieved a lower MSE. Furthermore, using the MSSSIM loss function led to the best SSIM and PSNR performance as expected. When comparing the results of DALL-E to other models on the UIED dataset, the performances are not quite as high as their results, where all three metrics are lower.

We also selected a few random images from the UIED dataset as shown in Figure 5 to explore qualitative results. One of the trends across all the DALL-E models is that they tend to produce blurrier images that appears to more fuzzy than both the input image and the reference image. One possible explanation for this is simply due to the encoder-decoder architecture: the encoder needs to compress the image into 1024 discrete tokens, which the decoder might have difficulty producing precise and sharp images with. It is not surprising that upscaling/decompressing an image from discrete tokens would give blurrier images. Even though the final image might not be extremely precise, the decoder is able to recover every components of the image without any extra artifacts.

The DALL-E models trained on L1 and MSE performed worse than the model trained on MSSSIM. In both the training and the evaluation sets, it appeared as if the model barely modified the input image colorwise, and the generated image is only blurrier than the input. As a result, the difference between the images generated by these two models look starkly different from the reference image. On the other hand, while not perfect, it appears that the DALL-E model trained on MSSSIM does make some changes to the input image in terms of its colors.

One limitation of the DALL-E model is the fact that the generated image draws heavily on the colors of the input image even if it needs to generate an image with dramatically different colors from the input. This is especially apparent in the 3rd and 4th rows in Figure 5. Since the input images are overwhelmingly blue or green, the generated images are also the same colors without significant changes. Even

though DALL-E MSSSIM attempted to adjust the colors, they are still not the same colors as the reference images.

5. Discussion

5.1. Two-Stage Network

In general, we observe fairly decent results using this simple model approach. Being limited in computational resources, we appreciated the simplicity that taking a two stage approach brought. We found that the colorization portion of the model was able to enhance a perfect version of the grayscale image to a very competitive level, while the grayscale enhancement network produced promising quantitative results for its own task, but worked less well when combined with colorization. An approach where both tasks are trained end-to-end, with an auxiliary loss guiding the grayscale enhancement and another loss guiding the colorization portion could be interesting to explore.

However, we are concerned about the model's ability to generalize to other datasets, and especially to the real world. Due to the lack of large amounts of ground truth data, we have relied heavily on small datasets for training and testing, which makes them easy to overfit on. Even though we observe good results on the unseen UIED dataset, it is still not a significant distribution shift from the EUVP dataset it was trained on, as both largely contain images dominated by blue or green, with close up shots of fish or coral dominating much of each dataset–this limits the practical use for neural underwater image enhancement models that are only trained on publicly available datasets. For instance, the datasets do not contain, or contain very few images of muddy waters or very deep oceans with artificial lighting.

5.2. DALL-E

One strength of using DALL-E is its extensive pretraining. Theoretically, this means that the model should be strong when fine-tuning to downstream tasks, similar to what happens in other well-known Transformer models that use large-scale pretraining. Because of this, it can also be easily fine-tuned to adapt to tasks like underwater image enhancement, which means that downstream models can benefit from a shared feature representation without the need to carefully design and architect new models. However, the DALL-E model in our experiment failed to produce meaningful results for the task. This could be due to a number of reasons, such as the complexity of the model for fine-tuning, or that the loss is not a strong enough signal for the model to learn on.

Although we did not explore the text aspect of DALL-E in this paper, it would be a promising next step. Since using text was a substantial part of the DALL-E's pre-training procedure and it has been shown that the text can influence the generated image's colors and texture [18], it is possible that input prompts like "remove the water in the image" could have improved the quality of the generated image.

One weakness of this model architecture is its size: it contains almost 10 million parameters, which is magnitudes bigger than other models used in this task. As a result, both training and inference can be much slower than other models, and it is necessary to use GPUs in order to run the model in a reasonable time frame. However, some may prioritize the speed of the model if they want to deploy the model in real time as they are underwater, which motivated researchers like Islam (2020) [12] to develop models that are able to process more than 7.9 frames/images per second on GPUs. Another computational limitation is the amount of memory it takes to fit all the parameters in the model. Due to its size, it would be difficult to run the model on smaller chips that are often part of underwater devices.

6. Conclusion

In this project, we explored deep learning approaches to underwater image enhancement, and achieved competitive results using a two-stage system when observing qualitative and quantitative samples. While the generative transformer model was not as performant, we gained a lot of interesting insights into how these large pretrained models function, and how they can be applied as bases for other downstream tasks.

Acknowledgments

We would like to thank Professor Jia Deng for teaching us many computer vision topics throughout the semester. We would also like to thank our TA Zeyu Ma for giving us feedback and guidance on our project proposal as well as answering our questions throughout the semester.

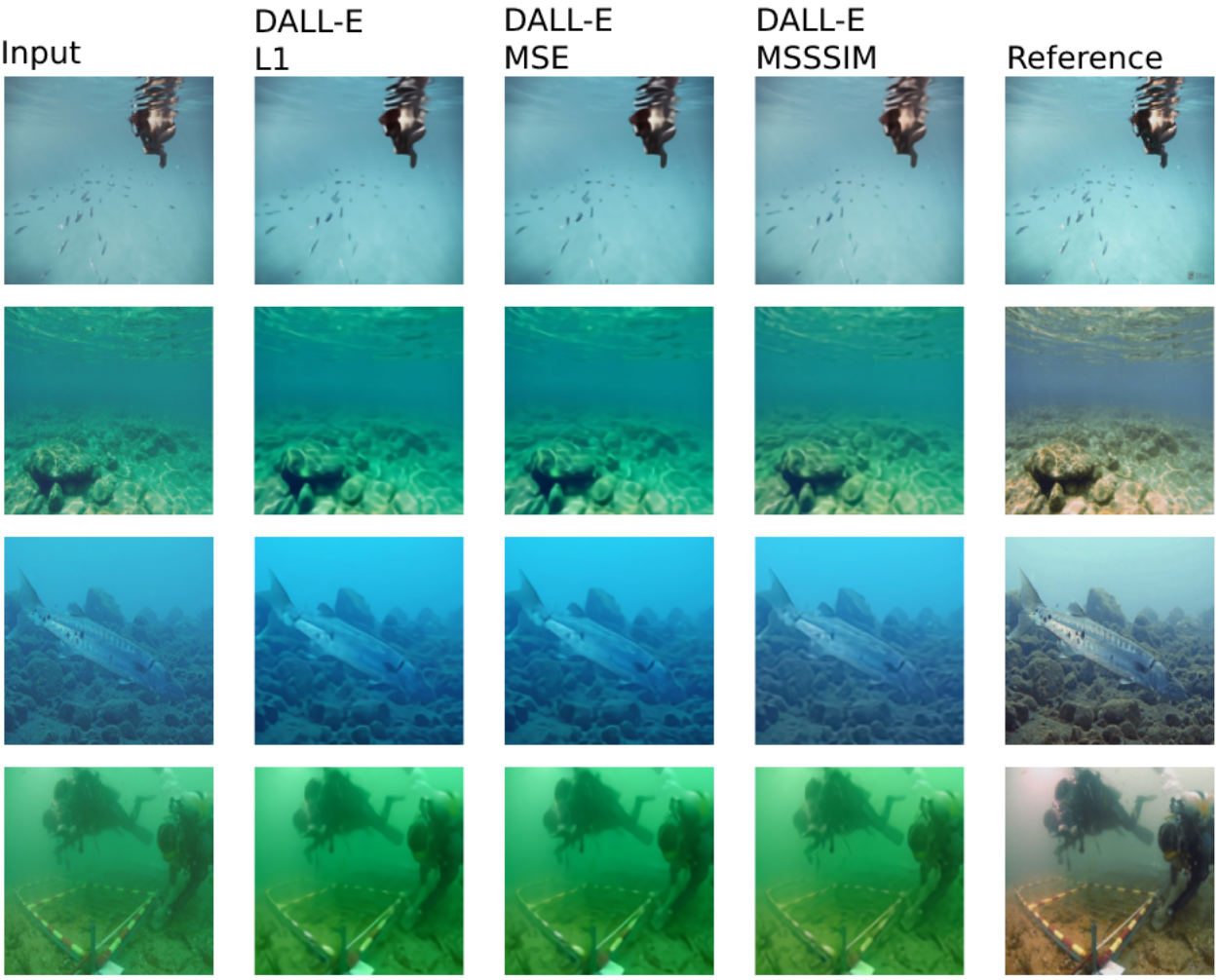

Figure 5. Examples of the generated images from the UIED (evaluation) dataset. From left to right, we show the input/raw image, the image generated by the three types of DALL-E models, and the reference image.

References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2016.

[2] Saeed Anwar and Chongyi Li. Diving deeper into underwater image enhancement: A survey, 2019.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[4] Keming Cao, Yan-Tsung Peng, and Pamela C. Cosman. Underwater image restoration using deep networks to estimate background light and scene depth. In *2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 1–4, 2018.

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer, 2021.

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Prafulla Dhariwal and Alex Nichol. Diffusion models beat

gans on image synthesis, 2021.

[9] Cameron Fabbri, Md Jahidul Islam, and Junaed Sattar. Enhancing underwater imagery using generative adversarial networks, 2018.

[10] Yecai Guo, Hanyu Li, and Peixian Zhuang. Underwater image enhancement using a multiscale dense generative adversarial network. *IEEE Journal of Oceanic Engineering*, 45(3):862–870, 2020.

[11] Md Jahidul Islam, Youya Xia, and Junaed Sattar. Fast underwater image enhancement for improved visual perception, 2020.

[12] Md Jahidul Islam, Youya Xia, and Junaed Sattar. Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters*, 5(2):3227–3234, 2020.

[13] Manoj Kumar, Dirk Weissenborn, and Nal Kalchbrenner. Colorization transformer, 2021.

[14] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2017.

[15] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond, 2019.

[16] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

[17] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[18] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[20] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations, 2020.

[21] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis, 2021.

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[23] Nan Wang, Yabin Zhou, Fenglei Han, Haitao Zhu, and Jingzheng Yao. Uwgan: Underwater gan for real-world underwater color restoration and dehazing, 2021.

[24] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning, 2019.

[25] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization, 2016.