

Investigations into Ultra-Low-Power Underwater Imaging

by

Nazish Naeem

BS., Lahore University of Management Sciences (2021)

Submitted to the Program in Media Arts and Sciences, School of
Architecture and Planning

in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2023

© 2023 Nazish Naeem. All rights reserved.

*The author hereby grants to MIT a nonexclusive, worldwide, irrevocable,
royalty-free license to exercise any and all rights under copyright,
including to reproduce, preserve, distribute and publicly display copies
of the thesis, or release the thesis under an open-access license.*

Authored by: Nazish Naeem

Program in Media Arts and Sciences

August 18, 2023

Certified by: Fadel Adib

Associate Professor

Thesis Supervisor

Accepted by: Joseph A. Paradiso

Academic Head, Program in Media Art and Sciences

Investigations into Ultra-Low-Power Underwater Imaging

by

Nazish Naeem

Submitted to the Program in Media Arts and Sciences, School of Architecture and
Planning

on August 18, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Imaging underwater environments is crucial to advancing our understanding of marine organisms, climate change, marine geology, aquaculture farming, and underwater archaeology. Despite significant advances in underwater imaging, scalable and long-term imaging of underwater environments is still an open problem. One of the main challenges in scalably imaging the ocean is that existing underwater cameras are too power-hungry for long-term observations. Recent work on ultra-low-power underwater imaging has shown that in-situ wireless underwater imaging is possible using fully submerged battery-free cameras and acoustic backscatter. Even though this is a promising advance, enabling truly useful ultra-low-power underwater imaging remains difficult due to many challenges and constraints including poor image quality (due to marine snow, hazing, and lighting conditions), limited energy, limited memory and computational power, and low bandwidth of the acoustic channel.

This thesis investigates the various challenges that efficient and ultra-low-power underwater imaging faces and offers directions for solving them. In particular, we first survey the various challenges of ultra-low-power underwater imaging. Subsequently, we offer three solutions for addressing these challenges. First, we propose a simple denoising/desnowing method for ultra-low-power underwater imaging that shows $\sim 2dB$ improvement in the quality of the images while reducing the memory consumption by $\sim 17x$ when compared to the state-of-the-art systems. Second, we perform ultra-low-power underwater edge inference that is $\sim 19x$ more memory efficient when compared to the baseline model with comparable accuracies. Then, we propose a solution for enabling ultra-low-power color imaging that is $\sim 10x$ less power-hungry than the state-of-the-art battery-free underwater imaging system. We conclude by offering a path to integrating these solutions into future end-to-end ultra-low-power underwater imaging systems.

Thesis Advisor:
Fadel Adib
Associate Professor

Investigations into Ultra-Low-Power Underwater Imaging

by
Nazish Naeem

This thesis has been reviewed and approved by the following committee members

Prof. Fadel Adib.....
Associate Professor
Program in Media Arts and Sciences, MIT

Prof. Hamed Haddadi
Faculty of Engineering
Department of Computing, Imperial College London

Prof. Nicholas Lane.....
Professor
Department of Computer Science & Technology, University of
Cambridge

To all the women in Pakistan who never got to live their dreams.

Acknowledgments

First, I would like to offer my sincere thanks to my advisor, Prof. Fadel Adib for his consistent support, expert guidance, and valuable advice. His optimism, dedication, and consistent belief in me have made this thesis possible. Additionally, I would like to present my sincere gratitude to my thesis readers, Prof. Hamed Haddadi and Prof. Nicholas Lane, for their valuable feedback and suggestions.

I am also grateful to my collaborators Tara, Weitung, and Waleed whose consistent assistance across various aspects of this thesis has been invaluable. I would also like to extend my gratitude to my amazing lab mates, Tara, Laura, Maisy, Jack, Saad, Waleed, Isaac, Osvy, Purui, Weitung, Ahmed, Unsoo, and Aline, for all the technical, professional, and personal support. I would also like to offer my special thanks to our group admin, Aimee, for taking care of all the logistics and making sure we celebrate Halloween.

I am deeply grateful to my parents, Naeem and Shazia, and my sisters, Ramish and Iqra, for their unwavering love, support, sacrifices, and prayers that have been instrumental in shaping my journey to this point. I extend my heartfelt appreciation to my mentors, Dr. Muhammad Tahir and Dr. Momin Uppal, for their invaluable guidance and unwavering encouragement. A special mention goes to Waleed, for his support through all my highs and lows during my time away from home. Additionally, I want to express my gratitude to Naima for her infectious positivity and the light that she brought into my life during my Masters. I would also like to mention my (All over the world senior) friends — Dilkasha, Azan, Mufeez, Aafaq, Taimoor, Soban, Sadia, and Usama — for all those (not so) monthly hands-on and listening to all my rants. Last, but not least, I would like to thank all the strong women in my life, special mention Sana, who showed me how to make your dreams come true.

The research is supported by the Office of Naval Research, the National Science Foundation, the Sloan Research Fellowship, the MIT Media Lab, and the MIT Sea Grant.

Contents

1	Introduction	17
2	Desnowing Underwater Images	21
2.1	Related Work	21
2.2	The Problem with Past Approaches	22
2.3	A Low-Complexity Solution for Desnowing Underwater Images Method	23
2.4	Implementation and Evaluation	26
2.4.1	Dataset Generation	27
2.4.2	Baselines	27
2.4.3	Evaluation Metrics	28
2.5	Results	29
2.5.1	Qualitative Results	29
2.5.2	Quantitative Results	29
2.6	Conclusion	31
3	Edge Inference on Ultra-Low-Power Underwater Imaging Systems	33
3.1	Introduction	33
3.2	Related Work	34
3.2.1	Edge Machine Learning	34
3.2.2	Visual Wake Word	35
3.3	Edge Inference Deployment on an Ultra-Low-Power Underwater Camera	35
3.3.1	Baseline Model	36
3.3.2	Neural Architecture Search (NAS)	37

3.3.3	Fine Tuning	37
3.3.4	Quantization	38
3.3.5	Deployment	38
3.4	Implementation and Evaluation	38
3.5	Results	39
3.5.1	Accuracy and Flash Memory Consumption	39
3.5.2	Peak RAM Usage	40
3.6	Conclusion	41
4	Ultra-Low-Power Underwater Color Imaging	43
4.1	Introduction	43
4.2	Related Work	46
4.2.1	Color Restoration	46
4.2.2	Colorization	47
4.3	Background	48
4.3.1	Underwater Imaging Model	48
4.3.2	Underwater Imaging Model For Grayscale Images	50
4.4	WaveColor - An Ultra-Low-Power Underwater Color Imaging Method	52
4.4.1	Distance Estimation	53
4.4.2	Dehazing	54
4.4.3	Attenuation Coefficient Estimation	57
4.4.4	Color Estimation	58
4.5	Implementation and Evaluation	59
4.5.1	Baselines	59
4.6	Results	60
4.6.1	Power Consumption	60
4.6.2	Reconstruction Results	61
4.7	Conclusion	63
5	Conclusion	65

List of Figures

2-1	Differencing Algorithm for Marine Snow Removal The figure shows a high-level flow of the algorithm. The camera captures multiple images of the underwater scene (4 in this example) as shown in A) and D). Then the mask of the marine snow particles is made using the difference between two consecutive images as shown in B) and E). The mask is used to remove the snow particles from the first frame, as shown in C) and F). C) represents the mask removed from the frame at t1 and F) shows the mask removed from frame t3. The images shown in C) and F) are used to inpaint the holes created in the image owing to the marine snow particles and reconstruct the denoised/desnowed underwater image as shown in G)	24
2-2	Generated Dataset The figure shows an example image from the generated dataset for evaluating the proposed method. A) shows the image of the coral reef model captured in clean water which serves as the ground truth image and B) shows the image of the same scene in water with plant soil (with floating particles similar to natural water bodies)	27
3-1	Baseline Model. A) shows the data preprocessing steps (grayscale conversion and downsampling), B) shows the architecture of the baseline neural network with MBCConv backbone and C) shows the MBCConv block.	37

3-2	PeakRAM Usage. The figure shows the RAM memory footprint of the machine learning network on the y-axis and the life cycle (operator) on the x-axis.	40
4-1	Underwater Imaging Scene. The figure shows an underwater scene in which an underwater camera is imaging a red, green, and blue cube. The cubes are illuminated by natural illumination (solid white arrows). A part of natural illumination reaches the object whereas some part of it is scattered by small particles in the water (dotted white arrows). The figure also shows that the light reflected from the object experiences some absorption (solid color arrows) and scattering (dotted color arrows) before reaching the camera. The captured camera image is shown in the top left corner of the figure.	48
4-2	WaveColor - Method Overview The figure shows an ultra-low-power grayscale underwater camera capturing an underwater scene. The underwater scene is illuminated by uniform natural illumination. A part of this light illuminates the underwater scene whereas some part of it is scattered by suspended particles in the water (also known as backscattering light). The underwater camera captures the underwater scene at one location and then moves by a distance Δd to capture another image. As the camera changes its location, the light reaching the camera changes and the camera captures a slightly different/attenuated grayscale image.	52
4-3	Veiling Effect in underwater Images. The figure demonstrates the hazing/veiling effect in the underwater images. A) shows the underwater images captured at different distances and the growing veiling effect with the distance. B) shows the same images after dehazing. . .	55

4-4	Average Pixel Value Curve - Before and After Dehazing. The figure shows the experimental average pixel values of a specific color before and after dehazing on the captured images at different distances. The blue curve shows the average pixel values before the dehazing. These images have both direct path and scattering light. Whereas, the purple curve shows the average pixel values after dehazing and thus these images only comprise the direct path light.	56
4-5	Power Consumption. The figure shows the comparison between the power consumption of the state-of-the-art battery-free underwater camera (blue curve) [3] and WaveColor(purple curve). The power-limited phase of these cameras corresponds to the time when the camera is actively imaging (the spikes in the curve). The flat (few microwatts) region corresponds to the time in which the camera backscatters the images.	61
4-6	Qualitative Analysis. The figure shows the qualitative results of the proposed method against the state-of-the-art colorization, and colorization and color restoration method. The first row shows the input grayscale image and the corresponding captured image. The second row shows the results of the three methods (Colorful Image Colorization [81], Colorful Image Colorization followed by UWCNN [8], and WaveColor)	62

List of Tables

2.1	Qualitative Analysis of the Proposed Algorithm Against the Baselines. The table shows the qualitative comparison of our proposed snow removal algorithm against the state-of-the-art denoising and desnowing methods.	29
2.2	Quantitative Analysis of the Proposed Algorithm Compared with the Baselines. The table compares the performance of the proposed algorithm against the baselines quantitatively. The table shows the PeakSNR(PSNR), Structural Similarity Index Measure(SSIM) and CIEDE 2000 (Color Distance) of the three methods.	30
2.3	Memory Consumption. The tables compares the memory consumption of the proposed method against the baselines algorithms. The table shows the number of parameters that need to be saved, in case of all three methods, in order to perform image cleaning on the input image and the size of the input image.	30
3.1	Accuracy and Memory Consumption. The table compares the model accuracy, number of parameters, and number of MAC operations of the baseline model, pruned model, and the final low-level compressed model.	39
4.1	Quantitative Analysis. The table compares the color difference (CIEDE2000) of the state-of-the-art colorization method, staet-of-the-art colorization followed by color correction method and WaveColor .	63

Chapter 1

Introduction

Underwater imaging is an important problem for many domains and scientific disciplines. Images of marine organisms, aquatic plants, ocean floors, and particulate organic carbon play a vital role in advancing our understanding of marine environments and their impact on global climate change [62, 58, 61]. Underwater imaging also plays a role in food production and food security; in particular, underwater cameras are used in aquaculture (seafood farming) to monitor different fish in order to detect diseases such as sea lice and monitor fish growth [17, 78]. Underwater imaging also enables the discovery of new marine species and helps us understand the impact of human activities on the marine ecosystem [57, 42].

Despite advances in underwater imaging, more than 95% of the ocean has never been explored or observed [29]. This is because we cannot perform in-situ large-scale underwater imaging. The key challenge that stands in the way is that existing methods for continuous underwater imaging either require tethering or bulky batteries for power and communication which limits their lifetime and scale of operation. Recently, there have been exciting advances in underwater imaging that have demonstrated the feasibility of doing underwater imaging, including marine life, plants, and underwater navigation tags, that are both battery-free and wireless [3]. However, while the early results are promising there is still a major leap that is required between these early feasibility results and the ability to do meaningful and useful ultra-low-power underwater imaging. Specifically, existing systems remain limited in their deployability,

efficiency, image quality, and end-to-end capabilities. To this end, *this thesis focuses on investigating a path toward sustainable ultra-low-power underwater imaging, its challenges, and an analysis of existing and potential solutions.*

Enabling ultra-low-power, efficient, sustainable, and useful underwater imaging requires addressing the following key challenges:

- **Image Quality:** Achieving high-quality underwater imaging is challenging for two main reasons. The first reason is that underwater environments introduce artifacts in the captured images. Specifically, the presence of suspended particles in water, a varied range of turbidity, and wavelength selective nature of the water introduce artifacts, like underwater backscatter, marine snow, color degradation, etc, in the images. Second, ultra-low-power underwater imaging sensors are low-resolution due to their limited power budget [31], adding an additional constraint to the quality of images. These not only degrade the quality of the images but also limit the usefulness of the images in different vision-based applications, discovery of species, disease detection, etc.
- **Energy Constraints:** In underwater sensing, one of the biggest challenges is charging or replacing the sensors' batteries, which is both expensive and labor intensive. Recent developments in the field of underwater backscatter [39, 25] have opened the door for battery-free underwater sensing and communication, eliminating the need for replacing batteries frequently. Recent work in underwater imaging [3] has made use of this technology to achieve battery-free and wireless underwater imaging by means of a remote transducer. However, these cameras still require a continuous source of energy to operate, which is obtained from a nearby sound source. In the absence of such a source, they become in-operational. Thus, they remain lacking a mechanism to enable long-term and resilient underwater imaging.
- **Memory and Computational Constraints:** Enabling underwater imaging based applications requires processing the captured images since they are not useful in their raw form. The biggest challenges that ultra-low-power imaging

platforms face, in processing these images, are the memory and computational constraints of the ultra-low-power edge processors on them. These ultra-low-power edge processors have a limited memory budget and computational power on them because of their energy-efficient embedded system design. For instance, the ultra-low power series of STM32 boards typically offers RAM of a few hundred of kilobytes and FLASH of a few megabytes [72]. This puts a constraint on both the amount of processing that can be done and the data that can be stored on the device adding computational and memory constraints on the imaging system.

- **Bandwidth Constraints:** Wireless underwater acoustic channel is inherently bandlimited and allows only a few kilobits per second (kbps) of data rates for communication. This not only slows down the communication process but also increases the energy consumption of the device. Specifically, in the past work [3] the battery-free underwater camera took 40 minutes to backscatter one grayscale image.

While each of these constraints is individually difficult, their combination makes the problem of ultra-low-power underwater imaging even more challenging. For example, ideally, we would like to be able to do edge compression before transmitting in order for us to enable low-latency imaging while operating within the limited bandwidth of underwater communication channels; however, more complex compression algorithms typically consume more energy and memory. Similarly, doing complex tasks, like recognition, monitoring, and segmentation (to further pre-process the data prior to transmission), are difficult not only because the computation, memory, and power are limited but also because the underwater imaging environments are very challenging to work with. Underwater environments change with changing light conditions, water quality, and region of imaging. Ideally, we would want to both adapt to such changing environments and remain within the resource budget. All of these make the problem of underwater imaging extremely rich as well as a difficult-to-solve problem.

This thesis explores and investigates a series of techniques in order to solve the aforementioned problems. In Chapter 2, we investigate the ways of removing artifacts from underwater images, problems with the existing approaches that make them unsuitable for ultra-low power imaging systems, and implementation and evaluation of a low-complexity solution for underwater image cleaning that is suitable for ultra-low power underwater imaging. This chapter also discusses how underwater artifact removal not only improves the quality of the images but in some cases also removes the unwanted information from the data reducing the amount of data to be communicated, benefiting both in terms of latency and overall energy consumption. Then, in Chapter 3, we talk about super-resolution as a technique to overcome the low resolution of the imaging sensor. This chapter investigates the super-resolution techniques that have been used in the past and how they can be helpful in terms of ultra-low-power underwater imaging. In Chapter 4, we investigate the possibility of colorization and color correction of the underwater images by means of processing the images based on the underwater imaging model and why existing methods fall short. Next, in Chapter 5, we investigate and discuss the possibility of using edge machine learning in an ultra-low-power underwater imaging system in order to solve one or many above-mentioned problems faced by such systems. This chapter also talks about the challenges that this technique faces when deploying in an underwater environment on an extremely resource-constrained platform.

Chapter 2

Desnowing Underwater Images

The presence of underwater artifacts lowers the quality of the underwater images which, as a result, reduces the usefulness of an imaging system. Removing artifacts, such as marine snow, turbidity, veiling effect, underwater backscatter, etc, from an underwater image is a well-known problem and is necessary for implementing vision-based tasks for robotics, environment monitoring, marine mammal detection, and species discovery. Enabling underwater image enhancement can increase the usefulness of the images and, in some cases, reduce the data to be stored and communicated by removing unwanted data from the image.

This chapter explores past work, investigates new techniques that follow the aforementioned criteria, and proposes a simple and effective desnowing algorithm.

2.1 Related Work

Artifacts in underwater images can be compared to noise and snow present in airborne images. Denoising and desnowing in-air images is an extensively studied problem, however, these models cannot be directly applied to underwater images since underwater images have inherent differences from in-air images. These disparities include color selectivity, veiling light, marine snow, and underwater backscatter. Furthermore, these artifacts are heavily influenced by water and lighting conditions, rendering the underwater imaging problem even more challenging [40]. With the emergence

of deep learning technologies, there have been several efforts to denoise and enhance these underwater images [53, 67, 30, 85]. But these deep learning-based techniques suffer from the lack of training datasets available for underwater images [40, 67]. Owing to the diverse underwater artifacts and our limited capacity to image the underwater world, it is hard to obtain the data and its ground truth for training any supervised network [40]. There have been efforts to synthesize training data, for underwater image processing, using either the underwater image model [67] or cGANs (cyclic generative adversarial networks) [83, 53], which are not true representatives of the real underwater images. Furthermore, deep learning-based underwater image denoising and enhancing methods remain very expensive in terms of memory footprint and computation cost, which prevents deploying them on a ultra-low-power edge device [50].

Apart from the deep learning-based solutions, researchers have also worked on removing underwater artifacts using different mathematical and image processing techniques like Laplacian pyramids [7], hierarchical processing [23], foreground and background separation [16], and even using special purpose cameras including RGBD and light field cameras [5, 70]. These methods are computationally expensive and water quality dependent thus they are not deployable on ultra-low-power cameras and are not generalizable for all underwater imaging environments.

2.2 The Problem with Past Approaches

Although there has been development on removing artifacts from underwater images, there has been little investigation into the problem of artifact removal from underwater images on an ultra-low-power camera. In order to realize underwater artifact removal on an ultra-low-power camera for long-term and sustainable imaging, the past approaches cannot be directly used because of the following reasons:

- These algorithms are computationally expensive. Specifically, past approaches either rely on machine learning or complex mathematical modeling, necessitating a large number of computations. In particular, these algorithms demand the

utilization of GPUs and sophisticated processors [53, 14, 55], such as Intel Xeon E5-1630 [53], to facilitate artifact removal in underwater images. Conversely, ultra-low power underwater cameras are equipped with only a few megabytes of Flash memory and a few hundred kilobytes of RAM [72], accompanied by a 32-bit Arm Cortex processor. While these ultra-low-power microcontrollers enable edge processing, they are incapable of meeting the computational requirements of previous approaches.

- Secondly, underwater imaging environments exhibit a high degree of diversity. These environments are heavily influenced by factors such as variations in lighting conditions, water types, and operational regions [4]. However, there is limited availability of representative data encompassing every underwater environment. This limitation arises due to our constrained ability to capture accurate imagery of the underwater world and acquire ground truth data for training machine learning models. Specifically, machine learning models trained under one set of conditions (lighting, water type, region, etc.) lack robustness when applied to different water types [6]. Recent work [16] has also proposed to calibrate the image enhancement method with changing water conditions but these approaches prove insubstantial for constantly changing underwater environments. Therefore, when designing an underwater image processing algorithm, it is important to consider these dynamic conditions and develop a robust algorithm capable of accommodating environmental changes.

2.3 A Low-Complexity Solution for Desnowing Underwater Images Method

Given that existing methods for denoising underwater images are complex for our power, memory, and compute constraints, we ask: *Is it possible to develop a low-complexity algorithm that works sufficiently well for these environments?* Our intuition for coming up with an algorithm stems from the problem itself.

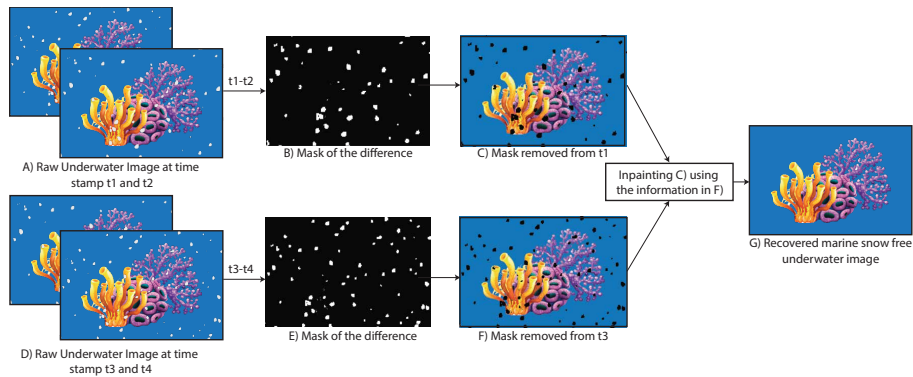


Figure 2-1: **Differencing Algorithm for Marine Snow Removal** The figure shows a high-level flow of the algorithm. The camera captures multiple images of the underwater scene (4 in this example) as shown in A) and D). Then the mask of the marine snow particles is made using the difference between two consecutive images as shown in B) and E). The mask is used to remove the snow particles from the first frame, as shown in C) and F). C) represents the mask removed from the frame at t_1 and F) shows the mask removed from frame t_3 . The images shown in C) and F) are used to inpaint the holes created in the image owing to the marine snow particles and reconstruct the denoised/desnowed underwater image as shown in G)

Though marine snow and underwater backscatter are two different phenomena, the goal of our algorithm is to design a general image processing technique that is capable of removing these artifacts without making the cameras task-specific and keeping the memory and energy budget in consideration. The design of our low-complexity algorithm relies on the observation that the majority of the artifacts in underwater images are due to suspended particles in water, be it marine snow or underwater backscatter. These little particles are not stationary and the motion of these particles is relatively rapid as compared to the motion of many marine animals. Thus, in principle, one could detect and/or eliminate these artifacts by implementing a simple differencing technique.

Mathematically, the underwater images with such artifacts can be modeled as the sum of the background B and time-varying marine snow particles $S(t)$.

$$I_t = B + S(t)$$

Our algorithm exploits this observation by taking multiple images and processing them onboard to remove these rapidly moving particles from the scene. Specifically,

we capture multiple underwater images with some time interval between each image as shown in Fig 2-1A and D. Since these images are captured at different time instances, randomly moving snow particles appear at different positions in each of the captured images. Now, we take the difference between two consecutive images i.e, for two images I_{t_1} and I_{t_2} taken at time instance t_1 and t_2 respectively:

$$I_{diff} = I_{t_1} - I_{t_2}$$

This is essentially the change in the first image with respect to the other image. This difference is, most of the time, rapidly moving particles. Next, we make a mask of these pixels fig. 2-1B and remove them from the original image as shown in fig. 2-1C. This not only removes the snow particles but also creates holes in the picture. Now we take the third and fourth images and perform the same set of operations as above, shown in fig. 2-1D, E, and F. Now we have two processed images -without snow and some missing background - as shown in fig. 2-1 C and F. To reconstruct the desnowed image, we use information from both images to fill in the missing parts of the other image, shown in fig. 2-1G. As a result of these operations, we have reconstructed a marine snow-free underwater image.

The algorithm of the proposed method is shown in Algorithm 1, where $I(t)$ is the captured image at time instant t , τ represents the number of unknown pixels in the image that needs to be recovered, τ_{th} represents the unknown pixel threshold, S_{mask} is the mask for the snow in the underwater image, R represents the recovered image and I_{diff} is the difference between two consecutive images. S_{mask} , R and I_{diff} are the same size as the captured image, and S_{mask} and R are initialized with 1 and ∞ respectively. In the beginning, the camera captures a second image, $I(t + 1)$, and we use this image to take the difference between the two consecutive frames and save it to I_{diff} . The difference in these frames helps in constructing the snow mask S_{mask} . For the unknown pixels (i.e., where $I_{diff} \neq 0$), we assign a value of ∞ to S_{mask} . As a result, S_{mask} equals 1 for known pixel values and ∞ otherwise. We use this mask to remove the unknown pixels from $I(t)$ and save it in I_{mask} . Now this masked image is

Algorithm 1 Algorithm for a Simple Difference based Underwater Image Desnowing Method

Captured Image at time t , $I(t)$
Number of unknown pixel values τ
Minimum number of unknown pixel values τ_{th}
Snow mask S_{mask}
Difference of Images I_{diff}
Recovered Image R
Initialize:
 $\tau = size(I(t + 1))$, $R = \text{array of } \infty \text{ of same size as } I(t)$,
 $S_{mask} = \text{array of } 1s \text{ of same size as } I(t)$
while $\tau \geq \tau_{th}$ **do**
 Capture $I(t + 1)$
 $I_{diff} = I(t) - I(t + 1)$
 $S_{mask}(i) = \infty$ where $I_{diff}(i) \neq 0$
 $I_{mask} = I(t) \cdot S_{mask}$
 $R(i) = I_{mask}(i)$ where $R(i) = \infty$
 $\tau \leftarrow$ Number of ∞ in R
 $t \leftarrow t + 1$
 $S_{mask} = \text{array of } 1s \text{ of same size as } I(t)$
end while

compared with the recovered image R to add in the known pixel values and then τ and t are updated and the process is repeated until $\tau < \tau_{th}$. Ideally, this threshold should be zero; however, if enough pixels are recovered then it becomes more effective and efficient to set this threshold to a small non-zero integer value and estimate the remaining unknown pixel values from the neighboring pixels than to recapture a new image.

2.4 Implementation and Evaluation

To study the effectiveness of our proposed algorithm, we evaluated its performance in a real underwater environment. We also implemented two state-of-the-art baseline algorithms (described below) for desnowing [14] and denoising [55] to compare the performance of our proposed algorithm. All the computation of the proposed algorithm was done in MATLAB and the images were collected using a GoPro camera.

2.4.1 Dataset Generation

To conduct a quantitative evaluation of the proposed algorithm’s performance, it is necessary to have ground truth images that are devoid of snow, in addition to the noisy images. However, due to the lack of existing datasets, we conducted a data collection procedure as outlined below. First, we placed an object (a coral reef model) inside a plastic tank filled with clean water to obtain artifact-free images. Subsequently, a camera was submerged in the water, and multiple images were captured from various angles. These images serve as the ground truth for our quantitative evaluation. Following that, we introduced underwater plant soil into the water and captured additional images while ensuring the camera remained stationary. An example of the collected images is shown in Figure 2-2. Figure 2-2A) displays the clean water image, representing the ground truth, while Figure 2-2B) illustrates the image after the addition of plant soil, representing the image with marine snow.

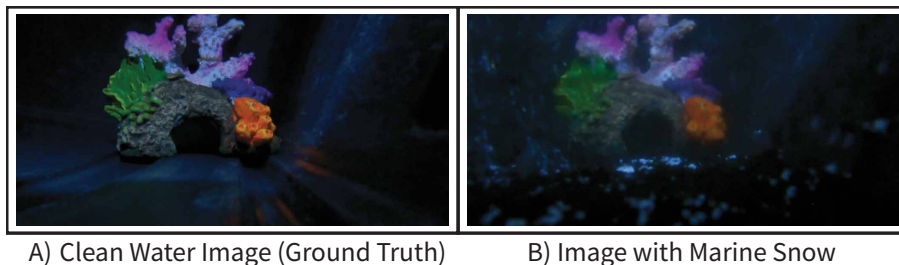


Figure 2-2: **Generated Dataset** The figure shows an example image from the generated dataset for evaluating the proposed method. A) shows the image of the coral reef model captured in clean water which serves as the ground truth image and B) shows the image of the same scene in water with plant soil (with floating particles similar to natural water bodies)

2.4.2 Baselines

We compare the performance of our algorithm against the following baselines:

- *All Snow Removal* [14] is a state-of-the-art snow removal method for in-air images that claim to remove all sizes of snow particles along with snow streaks in an image. It uses a hierarchical approach to remove all sizes and directions of snow particles from an image. Specifically, it uses Dual-Tree Complex

Wavelet Transform (DTCWT) [68] representation of an image and processes each level of the transform separately to desnow the image before combining them back. Each level passes through an image reconstruction neural network with Res2Net [24] backbone with 6.99M parameters. The results in the paper show that this method outperforms state-of-the-art desnowing networks with PSNR (Peak SNR) of 31.54 and SSIM (Structural Similarity Index) of 0.95 on Snow100K dataset [56].

- *Invertible Denoising Network (InvDN)* [55] is a state-of-the-art denoising network that uses an invertible neural network [9, 37], with 2.64M parameters, for removing noise from noisy images. This method uses invertible neural networks because of their lightweight, information-lossless, and memory-saving nature. InvDN transforms a noisy input image into a low-resolution clean image and latent representation containing noise. To denoise the image, InvDN replaces the noisy latent representation by sampling from a prior distribution of clean images before the backward pass. The results show that this method outperforms state-of-the-art denoising networks with PSNR of 39.28 and SSIM of 0.955 on SIDD dataset [2].

2.4.3 Evaluation Metrics

To evaluate the performance of the proposed method we use the following metrics:

1. **Peak Signal-to-Noise Ratio (PSNR):** PSNR is the ratio of the maximum possible value of the signal to the distorting signal (noise). For signals, such as images, with a wide range of values, PSNR helps in quantifying the quality of the signals.
2. **Structural Similarity Index Measure (SSIM):** SSIM is a quantitative measure of the perceived quality of the image. It is used to measure the structural similarity between the two images.
3. **Color Difference (CIEDE 2000):** CIEDE 2000 [69] is a color difference

formula developed in 2001. This formula quantifies the difference between the perceived color of two images.

2.5 Results

2.5.1 Qualitative Results

Table 2.1 shows the qualitative results of our algorithm against the baselines. The results show that our algorithm not only succeeds in snow removal but it also outperforms the other two algorithms. Though these are preliminary evaluations, these results show that it is feasible to design lightweight algorithms that can be deployed at the edge to perform effective image denoising.





Raw Underwater Image with Marine Snow	SOTA Denoising Method [55]
	
SOTA Desnowing Method [14]	Differencing based Marine Snow Removal (Our)
	

Table 2.1: **Qualitative Analysis of the Proposed Algorithm Against the Baselines.** The table shows the qualitative comparison of our proposed snow removal algorithm against the state-of-the-art denoising and desnowing methods.

2.5.2 Quantitative Results

Table 2.2 shows the PSNR, SSIM, and CIEDE 2000 of the baselines and our proposed method. Our proposed algorithm shows a higher PSNR and SSIM, and a lower

Method	PSNR	SSIM	CIEDE 2000
SOTA Desnowing Method [14]	19.0373	0.5668	9.5803
SOTA Denoising Method [55]	20.7097	0.5703	8.4569
Differencing based Marine Snow Removal (Ours)	22.8068	0.6419	6.5298

Table 2.2: **Quantitative Analysis of the Proposed Algorithm Compared with the Baselines.** The table compares the performance of the proposed algorithm against the baselines quantitatively. The table shows the PeakSNR(PSNR), Structural Similarity Index Measure(SSIM) and CIEDE 2000 (Color Distance) of the three methods.

	Number of Parameters	Input Size
SOTA Desnowing Method [14]	6.99M	<i>Input Image Size</i> (80kB)
SOTA Denoising Method [55]	2.64M	<i>Input Image Size</i> (80kB)
Differencing based Marine Snow Removal (Ours)	0	$2 * \text{Input Image Size}$ (160kB)

Table 2.3: **Memory Consumption.** The tables compares the memory consumption of the proposed method against the baselines algorithms. The table shows the number of parameters that need to be saved, in case of all three methods, in order to perform image cleaning on the input image and the size of the input image.

CIEDE 2000 as compared to the baselines. These improvements indicate the enhanced performance of our proposed desnowing method in comparison to the baselines. It is important to note that although the results show improvement, the absolute values of PSNR and SSIM are not exceedingly high. This can be attributed to the alteration of water properties by the introduction of underwater plant soil, which consequently introduces attenuation, scattering (as explained in Chapter 3), and small particles that contribute to the snow in the water. As our proposed algorithm focuses on simple desnowing, it is the relative change in evaluation metrics that holds significance, rather than the absolute values.

Table 2.3 compares the memory consumption of our proposed method with the baselines. The first row shows how many parameters each algorithm needs to save in memory. Our method doesn't require any parameters, while the baselines and other algorithms need to save millions of parameters in the memory. Similarly, the second row represents the size of the input that a method requires for processing. The

baselines only require one input image whereas our proposed method require two input images at single time instant. Specifically, for an ultra-low-power underwater camera, the baselines require 80kB of input data whereas our proposed method requires to save 160kB of input data at a time in the camera. Comparing both rows show that our proposed algorithm is much more memory-efficient compared to the state-of-the-art image desnowing and denoising algorithms, using significantly less memory.

2.6 Conclusion

The proposed image desnowing algorithm shows an average PSNR improvement of 2dB in the quality of the image while decreasing the memory consumption by a significant amount (eliminating millions of parameters). This shows that a simple effective algorithm can outperform advanced state-of-the-art mechanisms that have been developed for in-air processing. While this does not mean that future ML-based methods that are tailored to underwater environments would not perform better than the simple differencing method, it suggests that a low-complexity algorithm may offer an ideal first step for enhancing underwater images (for potential use in in-situ recognition/tracking tasks).

Chapter 3

Edge Inference on Ultra-Low-Power Underwater Imaging Systems

3.1 Introduction

One of the key tasks in underwater imaging is to perform analysis on different kinds of fish including the discovery of new species, detection of diseases, immigration and reproduction patterns etc. One way of doing so is to do analysis on all the captured images which includes processing and sending all the images back to a remote receiver. However, this is undesirable because underwater communication remains a high-latency and high-energy component of wireless underwater cameras [3]. Ideally, it would be more desirable to perform pre-processing locally and transmit only data that must be processed at a central server.

In this chapter, we exploit a few observations to perform edge inference (and ML) to realize this preprocessing task. One key observation is that these images are often sparse in both space and time, i.e., not all captured images would necessarily have a fish for analysis. Therefore a natural first step in such systems is to identify if there is a fish in the image before processing and sending back the data. This not only makes the system efficient but also makes it more accurate by removing false positives from the analysis. Object detection is a classic computer vision task that is realized via machine learning which makes it both memory and compute-intensive.

Recently, there has been extensive research on enabling machine learning on resource-constrained devices also known as edge machine learning. The rest of this chapter investigates the past approaches in similar domains and implements the first fish detection method on an ultra-low-power microcontroller.

3.2 Related Work

3.2.1 Edge Machine Learning

Edge machine learning enables the execution of machine learning processes on edge devices. Generally, machine learning models are large in size and necessitate servers with high computational power for running inferences. Earlier methods for running ML models on data collected from edge devices like smartphones required the edge device to transmit its data to the cloud for processing. More recently, researchers have dedicated significant effort to exploring ways of reducing machine learning model sizes, the number of operations, and quantization levels, among other factors, in order to enable machine learning inference on resource-constrained edge devices. Various model compression techniques, including knowledge distillation [32], neural architecture search [22], pruning [27], and quantization [26], have emerged that are capable of reducing model sizes from gigabytes to a few hundreds of kilobytes [51, 49]. Alongside model compression, it is equally crucial to consider RAM usage. Specifically, the network layers should be designed such that the activations in a layer fit within the RAM capacity of the edge device. Recent work has proposed and implemented patch-based inference [49] to mitigate the high memory usage during inference in the initial layers of CNN, thus reducing the peak RAM usage. Similarly, recent work has demonstrated the feasibility of training on microcontrollers [52] through Quantization-Aware scaling of gradients and sparse weight updates

3.2.2 Visual Wake Word

“Wake Word” refers to a phrase that acts as a trigger for edge devices to start their operation. For instance, devices such as Amazon Echo and Apple Siri keep listening for their wake word (Alexa, and Hey Siri) locally in order to start recording and sending user’s requests to the cloud for computation and inference. Similarly, visual edge devices use visual wake words to detect the presence of someone or something in the image. There is extensive past work on wake word detection on an edge device. However, many of the early detection algorithms could not be directly deployed on ultra-low-power microcontrollers that have orders of magnitude less memory than a voice assistance device or a smartphone [44, 76]. Recent works [64, 84, 43, 49] have explored the design and implementation of wake word systems on ultra-low-power microcontrollers. However, these models are not designed for ultra-low-power underwater imaging applications and therefore do not meet the requirements of such systems.

3.3 Edge Inference Deployment on an Ultra-Low-Power Underwater Camera

Given that previous approaches to edge machine learning have been relatively compute-intensive and unsuitable for ultra-low-power underwater cameras, this section takes the first step towards achieving ultra-low-power underwater edge machine learning. Specifically, we focus on implementing and evaluating the deployment of a fish visual wake word (fishVWW) on an ultra-low-power microcontroller. Similar to other visual wake words, fishVWW activates the camera when it detects a fish in the captured image. Its primary purpose is to serve as a trigger, indicating to the underwater camera that the captured image contains a fish and is ready for analysis. The subsequent subsections provide details on the method.

3.3.1 Baseline Model

The first step in developing fishVWW is to train a baseline model. A baseline model is a standalone and simple machine learning model that can perform the task we want our final edge model to perform. Specifically, like every other approach e.g., once-for-all network [13], knowledge distillation [32], pruning [27], etc, we require a model that can act as a teacher model that not only helps in narrowing down the search space but also offers a starting point for model compression.

Architecture: We train a baseline fishVWW model with MBCConv block backbone [66, 51] shown in Fig. 3-1C. The baseline model comprises 15 MBCConv blocks. The last MBCConv block serves as the classification layer for our network that works as a pointwise convolution layer and outputs only two channels. The network architecture is shown in Fig 3-1B.

Dataset: The network was trained on the DeepFish dataset [65] with an 80-20 split ratio for training and validation.

Preprocessing: Since the training dataset images are different from the images captured by a low-power imaging sensor, we preprocess the data to make the images similar to the ones captured from an ultra-low-power imaging system. Since the lowest power CMOS sensor [31] is grayscale, we convert all the 3 channel color images in the DeepFish dataset into one channel grayscale images. Similarly, we also downsample the images from the dataset to match the resolution of the ultra-low-power imaging platform as shown in fig. 3-1A. Note that since ultra-low-power CMOS sensors can allow multiple levels of resolution, e.g., VGA, QVGA, and QQVA, we can downsample and resize the training dataset to match the resolution of the mode the camera is being operated in.

Training: The baseline fishVWW network is trained using an SGD optimizer and cosine warm-up learning rate scheduler over 400 epochs.

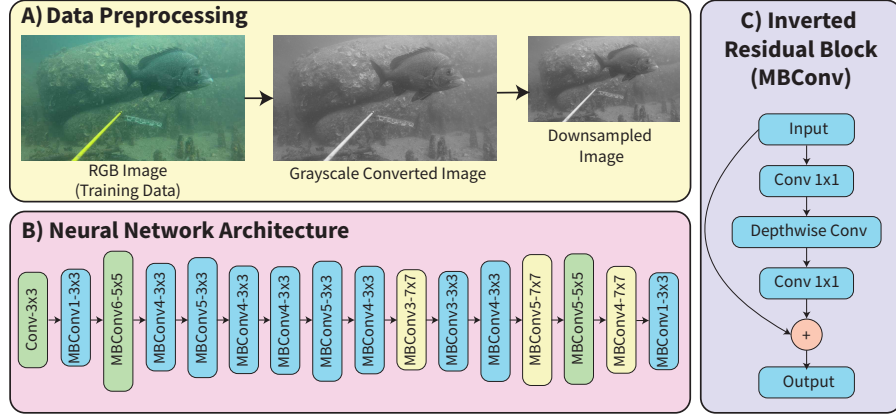


Figure 3-1: **Baseline Model.** A) shows the data preprocessing steps (grayscale conversion and downsampling), B) shows the architecture of the baseline neural network with MBConv backbone and C) shows the MBConv block.

3.3.2 Neural Architecture Search (NAS)

The next step after training a baseline model is to do a Neural Architecture Search (NAS). NAS is a process of finding the best architecture of a neural network from a search space that obeys certain constraints. There are different types of NAS depending on the application; in this implementation, we use automatic neural architecture search (autonas) [28]¹ with a constraint on the maximum number of parameters in the neural network as they are directly related to the FLASH memory of a microcontroller. Given a certain constraint and a baseline model, autoNAS gives a smaller network obeying the constraints.

3.3.3 Fine Tuning

After NAS, the next step is to fine-tune the searched model. AutoNAS searches for a smaller model that removes all the extra parameters, layers, and kernels. This reduced model has parameters of the old baseline model and hence does not give a good performance. At this stage, this new network needs to be fine-tuned (not trained from scratch). After fine-tuning, we have a smaller (searched) neural network model that is capable of performing fish detection.

¹AutoNAS is one of the well-known NAS techniques that involves a rigorous search of architectures given the certain constraints. We direct the reader to [28] for details of this method.

3.3.4 Quantization

After finetuning the searched model, we have a smaller neural network model that can perform fish detection. To further compress the model we quantize a model. Quantization is one of the most common, simple, and effective model compression techniques. There are multiple quantization techniques, for instance, k-means based quantization [26], linear quantization [36], per-channel weight quantization [60, 36, 59] etc from 8-bit integers to an extreme of 1-bit binary quantization [18]. We convert our model into an onnx model and then perform static quantization followed by a calibration step. Similar to NAS, quantization also impacts model performance and we perform another finetuning step to improve fish detection.

3.3.5 Deployment

After quantization, we have a model that is ready to be deployed. In order to deploy the model, we need to convert it into a low-level C code that can be interpreted by a microcontroller. Different microcontroller companies offer their own AI model converters. For instance, X-Cube-AI [74] and TinyEngine [51, 52, 48] are some of the known model conversion tools for STM’s ultra-low-power microcontrollers. TinyEngine was designed to make the inference process efficient in a microcontroller but recent updates in the X-Cube-AI have made the performance of both methods comparable. We use X-Cube-AI to convert the model into a low-level code (c-code). After C-code conversion, the model is ready to be deployed and tested on a microcontroller.

3.4 Implementation and Evaluation

Implementation: We trained a baseline pytorch model in python and then used OmniML [1] to run autoNAS with a constraint on the number of parameters to be less than equal to 160k. Next, we converted the pytorch model into an onnx model and performed static quantization and calibration. For deployment on a microcontroller,

we converted the onnx model into a low-level C code using X-Cube-AI and deployed it on an ultra-low-power microcontroller (STM32L476RG) [73]. It is worth mentioning here that researchers have used STM32L476RG [63] for battery-free underwater imaging.

Evaluation Metrics: We evaluated the performance of our method using model accuracy, number of floating point operations (FLOPs), and the peak RAM usage as evaluation metrics.

3.5 Results

3.5.1 Accuracy and Flash Memory Consumption

Table 3.1 shows the accuracy, number of parameters, and number of MACs (Multiply-accumulate operation) of the proposed model at different stages. The accuracy of the final compressed model shows a 0.7% drop from the original baseline model. However, the final compressed model is 18 times more memory efficient with 4 times less number of operations than the original baseline model. This shows that model compression can help us achieve similar results as a big machine learning model on a memory constraint device such as an ultra-low-power underwater camera.

	Model Accuracy	Number of Parameters (\propto FLASH Memory)	Number of MAC (multiply-accumulate) Operations
Baseline Model	99.7%	334k (\sim 2.67MB)	11.16M
Pruned Model (After AutoNAS)	99.4%	146k (\sim 1.17MB)	2.7M
Final Compressed (Low-Level) Model	99%	146k (\sim 146kB)	2.7M

Table 3.1: **Accuracy and Memory Consumption.** The table compares the model accuracy, number of parameters, and number of MAC operations of the baseline model, pruned model, and the final low-level compressed model.

3.5.2 Peak RAM Usage

In edge machine learning, peak RAM usage plays a crucial role in determining whether a machine learning model can be executed on an edge device. It represents the maximum memory required by a model layer during the inference process. While reducing the number of parameters in a neural network is often prioritized to ensure compatibility with microcontrollers, the activations of layers can also impose limitations on running the inference on an edge device.

Model parameters are typically stored in the FLASH memory of a microcontroller, while each inference step relies on the RAM of the device. Therefore, it is essential to assess the RAM requirements of each layer for successful inference on an edge device. To determine if a model can run on such a device, the peak RAM usage of the model should be lower than the available RAM memory.

Figure 3-2 illustrates the memory footprint of running the inference on a microcontroller using a neural network. The maximum memory footprint corresponds to the peak RAM usage. In our experiments, the proposed model exhibits a peak RAM usage of approximately 65.5kb, significantly lower than the 128kB RAM capacity of the ultra-low-power microcontroller (STM32L476RG) we utilized.

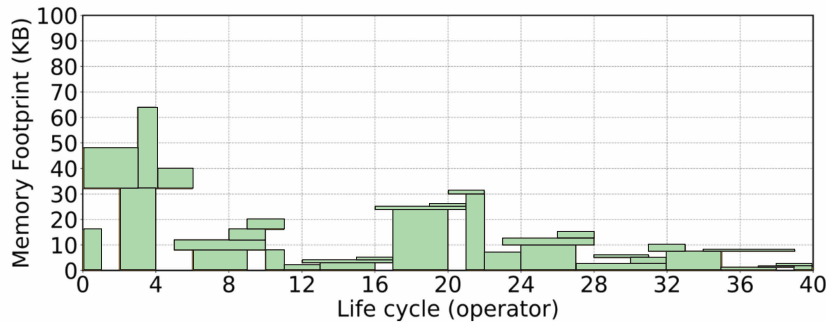


Figure 3-2: **PeakRAM Usage.** The figure shows the RAM memory footprint of the machine learning network on the y-axis and the life cycle (operator) on the x-axis.

3.6 Conclusion

Enabling fish detection on ultra-low-power underwater imaging systems has the potential to significantly enhance the efficiency of these platforms. The ability to detect fish serves as a trigger, allowing the cameras to selectively save and transmit only the important images. This approach greatly improves the memory and energy efficiency of the system. Additionally, ultra-low-power fish detection can also serve as a wake-up signal for more complex and resource-intensive operations, such as segmentation, tracking, etc on the ultra-low-power underwater cameras.

Implementing a fish visual wake word represents a crucial initial step towards enabling advanced tasks, including fish analysis, identification of new marine species, climate change monitoring, and more, on ultra-low-power edge devices. This development opens up a wide range of research opportunities for sustainable imaging of underwater environments in fields such as oceanography, climatology, marine biology, ecology, and beyond.

Chapter 4

Ultra-Low-Power Underwater Color Imaging

4.1 Introduction

Capturing *color* images of underwater environments is critical for various applications. Underwater color images can be used for the discovery of marine species, detecting diseases of animals, recognizing different fish in aquafarms, monitoring coral reef bleaching, and tracking geological processes (e.g., submarine volcanoes). Additionally, enabling ultra-low-power color imaging of underwater environments is important since it enables underwater cameras to last for long periods of time for sustainable, scalable, and long-term monitoring of marine ecosystems.

Unfortunately, capturing color images in underwater environments at low power is extremely difficult. Today, there are two approaches for capturing underwater color images, both of which are power-consuming in their own way: The first and most classical approach is to use an RGB image sensor; the lowest-power image sensor on the market is onsemi ARX3A0 [34], and it consumes $19mW$ for each image. The second, more recent approach [3], is to use a grayscale CMOS sensor and multi-color illumination. This approach was recently proposed and its idea is to illuminate the scene with a single LED (e.g., red, green, blue), capture separate grayscale images for each LED, and apply the grayscale images to the different RGB channels of a color

image. Interestingly, even though this approach requires LEDs, the average power consumption in power limited phase (i.e., when the LEDs are active) is $\sim 7.1mW$ ($1.1mW$ for the CMOS sensor and $1.9 - 8.2 mW$ for the LEDs), making it more power-efficient than using an off-the-shelf RGB image sensor. Thus, today, the most power-consuming component of this past method ($>90\%$ of power) comes from the need to capture color in images.

This chapter asks the following question: *is it possible to reconstruct color underwater images at even lower power and without LED illumination?* If we can do so, then we can drastically reduce the power consumption of underwater color images.

We propose WaveColor, a novel ultra-low power underwater color imaging method that is capable of capturing underwater color images without using active LED illumination or high power consuming RGB cameras. WaveColor exploits wavelength selectivity of the water in order to reconstruct the colors of the underwater scene. Specifically, the proposed method captures multiple grayscale images of an underwater scene at different distances and uses the decay in the colors of different parts of the scene to reconstruct the original color. Importantly, since many underwater cameras are deployed on (battery-free) floats or marine mammals, this approach would enable leveraging natural mobility to reconstruct color images while leveraging existing ultra-low-power grayscale CMOS sensors.

One might think that one could simply capture a grayscale image and then colorize the image using existing methods like those used in colorizing old films. However, doing this is not straightforward and would yield undesirable results (as we show in this chapter). Specifically, most of the colorization mechanisms are trained on videos taken above the surface, thus the deep-learning models are overfitting to above-air rather than underwater images. Moreover, the variable nature of underwater imaging environments makes this problem more challenging. Specifically, different lighting conditions, regions of imaging, distance of the camera and water types change the colors of an underwater object that appears in a captured image i.e. the same object will appear to have different colors in different water conditions making it difficult for any colorization method to colorize underwater images.

In contrast to these above methods, WaveColor is inherently designed for underwater image colorization, leveraging the wavelength-selectivity of the underwater environments. This method is particularly powerful for two reasons:

1. **Ultra-low-power:** First, it does not use any active illumination for imaging underwater environments i.e. the proposed method is capable of capturing color images using ultra-low power grayscale imaging sensors in the presence of ambient light. This significantly reduces the power consumption of the system.
2. **Generalizability:** Second, the method exploits the natural wavelength selectivity of the water in order to understand the water type and colorize the images. Unlike other colorization methods, this method does not depend on any training dataset and therefore is capable of working in practically any unseen underwater environments.

We implemented the first prototype of the method in MATLAB and conducted experiments in MIT SeaGrant in order to generate preliminary results. The results show that

1. WaveColor is capable of reducing power consumption of the state-of-the-art ultra-low-power color underwater imaging system [3] by 10x
2. The results also show WaveColor's capability of recovering real colors of underwater objects without any color degradation with an average color difference (ΔE) of 0.025.

In this Chapter, we talk in detail about past work on colorization and color correction, then we talk in detail about the physical underwater imaging model. After that, we describe the proposed underwater color imaging method, WaveColor, followed by some preliminary implementation and results.

4.2 Related Work

4.2.1 Color Restoration

With the increasing interest in underwater imaging, researchers have started exploring ways to remove underwater artifacts from the images. Specifically, researchers are actively exploring ways to enhance the colors of the underwater images to extract the true colors of the scene. With the advancements in technology, researchers have been exploring ways to use deep learning to extract the actual colors of an underwater scene. Past work [10] has explored the ways of removing the haze and blurring effect from a single underwater image. However, these techniques only remove the haze from the underwater images and fail to recover the true colors of the scene. Researchers have also explored ways of joint dehazing and color restoration of underwater images [47, 75, 21]. These methods include training neural networks with attention maps [75], using multiscale structures [75], channel-wise processing [46], and introducing new datasets [10] etc. Although these techniques help in recovering the colors and removing blurriness in the images, they fail to account for the effect of changing lights, water types, and distances. Recent work [47] shows that introducing light source and attenuation estimation in neural networks enhances the performance of the color restoration of underwater images. While all these advancements are encouraging, the main limitation of using deep-learning-based color reconstruction techniques is the lack of real-world datasets. Synthetic datasets [40] are not representative of the true underwater images and hence, they introduce an inherent bias in the neural network inferences. Some past works [83, 53] have explored the possibility of using cyclic GANs to learn the color images from the relatively vast dataset of in-air images. However, these images start adding bias from the in-air images (like clouds, sky, grass, etc) in the inference.

Researchers have also explored the possibility of using model-based techniques - which do not require any dataset for training - to reconstruct the actual colors from the environment. Some initial work in this domain [79] proposed to solve simple linear equations to reconstruct the absorbed colors in an underwater image. However,

in real underwater images, the imaging models are more complex than the proposed simplistic model. Researchers have taken into account these effects and proposed different underwater color restoration methods for both deep [71] and shallow waters [6]. These developments in enhancing underwater images are encouraging; however, none of the past approaches have explored the possibility of using grayscale images to reconstruct enhanced color underwater images.

4.2.2 Colorization

Colorization is the process of adding colors to a grayscale image. It is one of the classic computer vision problems that has been worked on for more than 20 years [80]. Past approaches [45, 82, 35, 77, 54] have discovered the possibility of using user-guided colorization techniques. These techniques use hints from the user in the form of scribbling [45, 82] some colors in the pixels or some example images [35, 77, 54] of the same or similar objects. Though these techniques show impressive performance, they require prior knowledge of the color to color the images. With the advancements in the field of machine learning, researchers have extensively started using neural networks to colorize the images [15, 11, 19, 33]. Although theoretically, this technique is also a kind of example-based reconstruction, these networks have shown remarkable performance on unseen data as well. Some recent works [82] have also explored the possibility of designing hint-based colorization neural networks that combine the benefits of both prior techniques. The process of colorization has evolved and improved over the past decades, however, these techniques fail to work on underwater images. Underwater images are very different, diverse, and complex as compared to airborne images, and therefore, these techniques cannot be directly used to colorize the captured underwater scenes. Moreover, the diversity of underwater images and our limited ability to obtain the training data make it difficult to train any new network that can colorize an underwater image.

4.3 Background

Before going into the details of the proposed color reconstruction algorithm, we briefly explain the underwater imaging model. This is important to understand phenomena like light attenuation and scattering, and how they affect underwater images. Specifically, we talk about underwater image formation and its mathematical modeling in both RGB and grayscale.

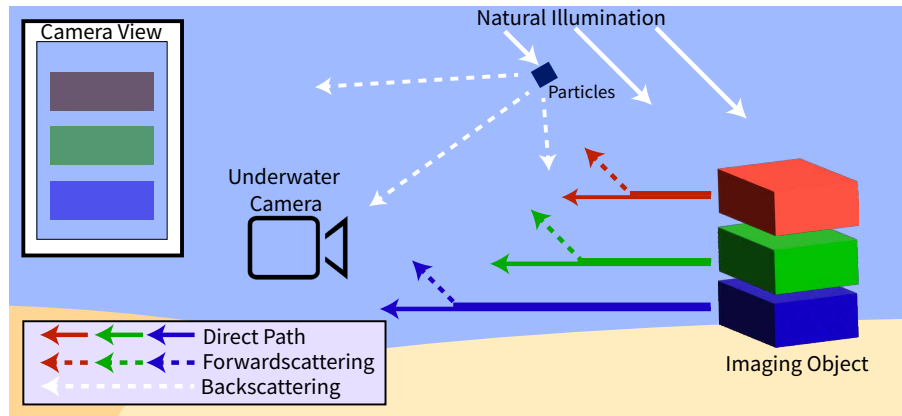


Figure 4-1: **Underwater Imaging Scene.** The figure shows an underwater scene in which an underwater camera is imaging a red, green, and blue cube. The cubes are illuminated by natural illumination (solid white arrows). A part of natural illumination reaches the object whereas some part of it is scattered by small particles in the water (dotted white arrows). The figure also shows that the light reflected from the object experiences some absorption (solid color arrows) and scattering (dotted color arrows) before reaching the camera. The captured camera image is shown in the top left corner of the figure.

4.3.1 Underwater Imaging Model

Underwater images, unlike in-air images, are complex to model. This complexity mainly comes from the water itself, the presence of different types and sizes of particles in the water, the light source, etc. Fig. 4-1 shows an underwater imaging environment in which a camera is capturing an underwater scene (three different color cubes). At the top left corner, we can also see the captured image. According to the Jaffe-McGlamery model, underwater images consist of three components 1. Direct path (solid red, green, and blue lines in fig. 4-1) 2. Backscattering (dotted white lines in

fig. 4-1) and 3. Forwardscattering(dotted red, green and blue lines in fig. 4-1 [38]. Direct path refers to the light that directly reflects from the object’s surface toward the camera. This direct path experiences attenuation depending on the wavelength of the light. Similarly, forwardscattering refers to the light scattered from the direct path due to the suspended particles in the water whereas backscattering, in the context of underwater imaging, refers to all other scattering of light captured in the camera that is not coming from the direct path. Both of these scattering components are also wavelength dependant. We can combine both the scattering lights as one component and, mathematically, an underwater image can be written as the sum of the direct path and all the scattering lights,

$$I_\lambda = D_\lambda + B_\lambda \quad (4.1)$$

where λ is the wavelength of the captured light, I_λ is the underwater image, D_λ is the direct signal, and B_λ is the backscattering light. B_λ includes both backscatter and forwardscatter and for simplicity, we would refer to it as scattering light. The direct signal D_λ has the information about the imaging scene whereas B_λ causes hazing and blurring in the image which is also known as the veiling effect.

Direct Path: As mentioned earlier, the direct path signal corresponds to the light that gets reflected from the imaging object and reaches the camera. This direct path signal gets attenuated because of the absorptive nature of the water. This absorption rate is wavelength dependent i.e., some wavelengths would get absorbed quicker than others. Similarly, this direct path’s absorption is also distance dependent i.e., as we move further away from the imaging object, more light gets absorbed. Mathematically, the direct path can be written as,

$$D_\lambda(d) = D_\lambda e^{-\beta_d(\lambda)d} \quad (4.2)$$

where D_λ is non-attenuated light, β_d is the attenuation coefficient which is wavelength dependent, and d is the distance between the imaging scene and the camera. The attenuation coefficient encapsulates the attenuation of light caused by both absorption

and scattering and therefore is often written as the sum of scattering and absorption coefficient [12].

Scattering Light: Similar to the direct path, the scattering light B_λ is also dependent on wavelength and distance and can be written as follows [6]:

$$B_\lambda(d) = B_\lambda^\infty(1 - e^{-\beta_b(\lambda)d}) \quad (4.3)$$

Where B_λ^∞ is the veiling light at an infinite distance from the object. And β_b is the decay coefficient of the scattering light which is a function of wavelength. Note that as the distance from the object increases, I_λ approaches B_λ (eq. 4.1) which means that as we move away from the object we see more and more scattering light and after some point, the scattering light starts dominating the direct path and therefore the captured intensity of the light starts increasing as we move further from the imaging scene¹.

All the derivation above assumes uniform illumination of the imaging scene. In the case of active imaging i.e., with an artificial light source, the imaging model becomes much more complex. This is because it needs to take into account the nonuniform illumination, angle of illumination, the shape of the spreading, etc [71].

4.3.2 Underwater Imaging Model For Grayscale Images

For a captured underwater color image, we have three color channels i.e., red, green, and blue. For a single channel, the captured intensity can be written as follows [4].

$$I_c = \frac{1}{\kappa} \int_{\lambda_1}^{\lambda_2} S_c(\lambda)\rho(\lambda)E(d, \lambda)e^{-\beta_d(\lambda)*d} d\lambda + \frac{1}{\kappa} \int_{\lambda_1}^{\lambda_2} S_c(\lambda)B_\lambda^\infty(1 - e^{-\beta_b(\lambda)*d}) d\lambda \quad (4.4)$$

where c is one of the three color channels, $S_c(\lambda)$ is the spectral sensitivity of the camera, $\rho(\lambda)$ is the reflectance of the underwater imaging scene, $E(d, \lambda)$ is the irradiance from the illumination, κ is a scalar that compensates for the exposure and offsets

¹We refer the readers to [38] for more details on the imaging model.

of the camera, and λ_1 and λ_2 correspond to the band of the electromagnetic spectrum of the channel. Although one channel does not correspond to one wavelength, we can still write the above equation as a sum of the direct path and scattering light,

$$I_c = D_c + B_c \quad (4.5)$$

Since the channel is no longer a single wavelength, the attenuation coefficients become a function of both channel (red, green, and blue) and distance. We refer readers to [4] for more details.

Now that we have the imaging model for a three-channel color image, let us derive the underwater imaging model for a monochromatic camera. A monochrome camera is a one-channel image sensor that captures the intensity of different colors that are incident on the imaging sensor. A grayscale image can also be represented as a weighted sum of red, green, and blue color channels. Given this information, mathematically a grayscale image can be written as,

$$I_{gray} = a_r R + a_g G + a_b B \quad (4.6)$$

Where a_r , a_g , and a_b are the weights of the different color channels and are mostly dependent on the camera's sensitivity. R , G and B are the three channel intensity and I_{gray} is the intensity of the grayscale image. From eq. 4.5, eq. 4.6 can be written as,

$$I_{gray} = a_r(D_r + B_r) + a_g(D_g + B_g) + a_b(D_b + B_b) \quad (4.7)$$

$$I_{gray} = a_r D_r + a_g D_g + a_b D_b + \textit{scattering terms} \quad (4.8)$$

We combine all the scattering terms in the equation above and get an imaging model for a grayscale imaging device. Equation 4.8 can further be expanded as follows,

$$I_{gray} = a_r(D_{airr}e^{-\beta_d(r,d)d}) + a_g(D_{airg}e^{-\beta_d(g,d)d}) + a_b*(D_{airb}e^{-\beta_d(b,d)d}) + \textit{scattering terms} \quad (4.9)$$

Where D_{airr} , D_{airg} and D_{airb} are the red, green, and blue channel pixel values respec-

tively, representing the actual colors of the underwater scene if they were captured in air. In order to do ultra-low-power underwater imaging using a grayscale camera, the goal is to estimate the values of D_{air_r} , D_{air_g} and D_{air_b} .

4.4 WaveColor - An Ultra-Low-Power Underwater Color Imaging Method

Now that we have the necessary background, in this section, we talk about an ultra-low-power underwater color imaging method.

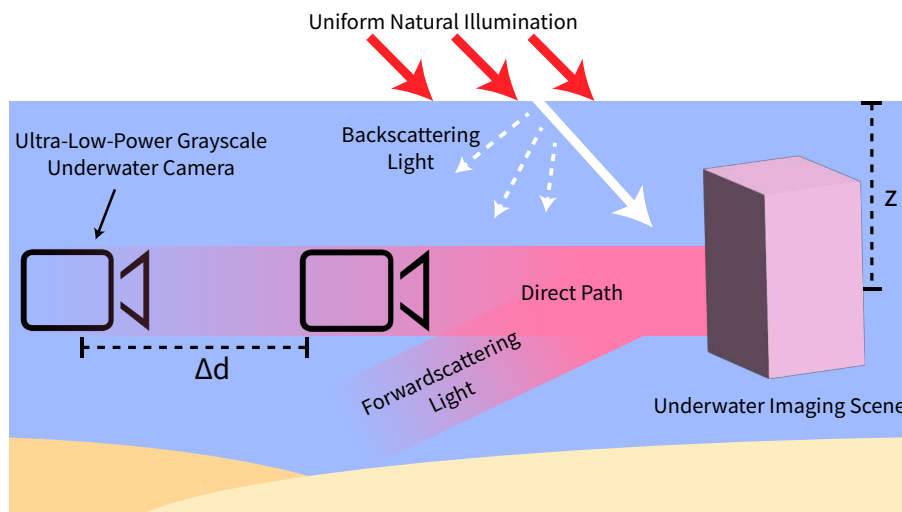


Figure 4-2: **WaveColor - Method Overview** The figure shows an ultra-low-power grayscale underwater camera capturing an underwater scene. The underwater scene is illuminated by uniform natural illumination. A part of this light illuminates the underwater scene whereas some part of it is scattered by suspended particles in the water (also known as backscattering light). The underwater camera captures the underwater scene at one location and then moves by a distance Δd to capture another image. As the camera changes its location, the light reaching the camera changes and the camera captures a slightly different/attenuated grayscale image.

The wavelength-selective nature of water degrades the colors in an underwater image and thus at different distances, the colors of the same underwater scene appear different. We exploit this observation and use the decay profiles of the colors of different parts of the underwater image, as the camera moves away from the scene, to reconstruct the real colors from multiple monochromatic images. Fig 4-2 shows an

underwater scene that is illuminated by natural light. The scene is being imaged using a grayscale camera that moves away from the scene by a distance Δd as it captures a grayscale image. As the camera moves away from the underwater scene, it captures the light attenuation and scattering patterns at different distances. WaveColor constitutes of the following steps

1. **Distance estimation:** This step involves estimating the distance of the camera from the underwater scene.
2. **Dehazing:** This step involves the removal of the estimated scattering component from the image. By removing the scattering, also known as haze in the image, we are only left with the direct path that has the information about the colors of the underwater scene.
3. **Attenuation coefficient estimation:** Next step is to estimate the decay coefficients of red, green, and blue channels of the color underwater image. This is crucial for the final color estimation process.
4. **Color estimation:** After estimating the decay coefficients, we estimate the colors of different parts of the underwater scene.

In the following sections, we describe these steps in detail.

4.4.1 Distance Estimation

The first step before starting the reconstruction process is to estimate the distance of an object from the camera. There has been extensive work on estimating the distance of the object from the camera [20, 41]. However, these methods cannot be used directly because these methods either rely on the prior information about the actual size of the object or are machine learning models that are trained for a certain class of objects. Instead, we leverage the motion of the camera in order to estimate the distance between the camera and the scene.

For a given object of width W that is D_1 distance away from the camera and has a pixel width of P_1 when captured at the given distance, the focal length of the

camera can be written as,

$$F = \frac{WP_1}{D_1} \Rightarrow D_1 = \frac{WP_1}{F} \quad (4.10)$$

Where the pixel width of an object is defined as the number of pixels in one dimension that contains the target object. Pixel width can easily be found by using any object detection and bounding box estimation algorithm. Similar to the previous equation, for another captured image at distance D_2 and pixel width W_2 , the equation above becomes

$$D_2 = \frac{WP_2}{F} \quad (4.11)$$

Subtracting equations 4.10 and 4.11,

$$\Delta d = \frac{W(P_1 - P_2)}{F} \quad (4.12)$$

Where $\Delta d = D_1 - D_2$ is the known distance between two consecutive camera positions and can easily be extracted using the time between two consecutive frames and the speed of the camera. By substituting eq. 4.10 in eq. 4.12 and rearranging, we get

$$D_1 = \frac{\Delta d P_1}{P_1 - P_2} \quad (4.13)$$

The equation above can be used to estimate the distance of the object from the camera given the pixel width of the object in two different frames and the change in distance between the given frames.

4.4.2 Dehazing

Now that we have estimated the distance of all the captured images, the next step is to remove any unwanted haze from the images. As mentioned in sec. 4.3, the suspended particles in water bodies are responsible for scattering of the light which shows as haze in the underwater images. This haze caused by scattering is also termed as the veiling effect [6]. Recall from eq. 4.8, the monochromatic image has a scattering term

that is responsible for the hazing effect in an underwater image which is both distance and wavelength dependent. Fig. 4-3A shows the grayscale underwater images of a

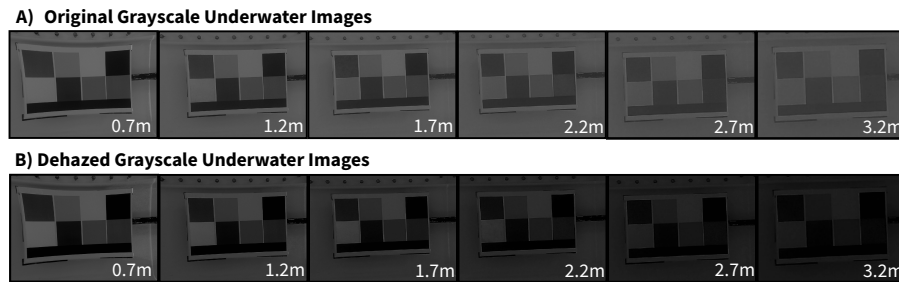


Figure 4-3: **Veiling Effect in underwater Images.** The figure demonstrates the hazing/veiling effect in the underwater images. A) shows the underwater images captured at different distances and the growing veiling effect with the distance. B) shows the same images after dehazing.

tag at different distances from the camera. It can be seen from the figure that as the distance between the camera and the tag increases, the veiling effect also increases. The same effect can also be seen quantitatively in fig 4-4. The blue curve shows the average pixel values of a certain color (blue in this example) against distance. The graph shows that instead of monotonically decreasing, the pixel intensity starts to increase after a certain point. This is because the scattering light intensity increases with increasing distance i.e., scattering starts dominating the direct path. Since the scattering does not have any useful information about the colors of the scene, the first step in the color reconstruction of grayscale underwater images is to remove the haze from the images. After removing the hazing effect i.e., the scattering from these images, we are only left with the attenuated direct path which makes it much easier to reconstruct the colors in the images. Fig 4-3B shows the dehazed underwater images at different distances. It can be seen from the figure that as the distance increases the color of the image (direct path) attenuates. This can also be seen quantitatively in fig 4-4 (purple curve) that the average pixel density of the dehazed images shows a monotonically decreasing trend as it mostly contains the direct path in it.

In order to remove the hazing effect, that is caused by scattering of light underwater, we need to know the light source and its illumination patterns. However, since WaveColor does not have its own light source and mostly relies on ambient light for

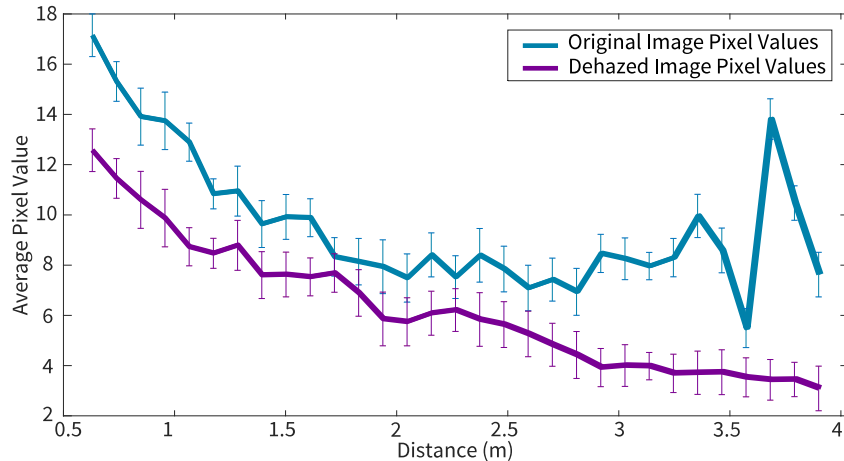


Figure 4-4: **Average Pixel Value Curve - Before and After Dehazing.** The figure shows the experimental average pixel values of a specific color before and after dehazing on the captured images at different distances. The blue curve shows the average pixel values before the dehazing. These images have both direct path and scattering light. Whereas, the purple curve shows the average pixel values after dehazing and thus these images only comprise the direct path light.

illumination, we can safely assume the illumination to be uniform. In the case of uniform illumination, it has been observed that at a given distance, the scattering effect is fairly uniform across all the pixels with similar depths. Past work [6] proposes depth-based scattering light removal using an RGBD camera. We take inspiration from the prior work and use the darkest pixels around the target object in order to estimate the scattering signal and remove it from the whole image. We can safely assume that for small objects in the image, the depth across all the pixels is similar. For testing the initial idea, we used known black pixels in order to remove the scattering signal. Although this method does not take into account the randomness of the scattering light, however, since the illumination is fairly uniform, this randomness does not greatly affect the pixel values. Moreover, any noise that might have been introduced by this randomness is taken into account in the color estimation process. (sec. 4.4.3).

4.4.3 Attenuation Coefficient Estimation

After dehazing the images, we remove the scattering part from the underwater images and the resultant images consist of direct paths only. Mathematically, eq 4.9 becomes,

$$I_{gray} = a_r D_r e^{(-\beta_d(r,d)*d)} + a_g D_g e^{(-\beta_d(g,d)*d)} + a_b D_b e^{(-\beta_d(b,d)*d)} \quad (4.14)$$

Recall that the goal of this method is to do underwater color imaging i.e., to estimate the values for D_r , D_g , and D_b . In order to do that, the next step is to find attenuation coefficients $\beta_d(r, d)$, $\beta_d(g, d)$ and $\beta_d(b, d)$ for the three color channels red, green and blue respectively. These attenuation coefficients are dependent on multiple things including the concentration and sizes of different particles in water, the distance between the camera and the imaging scene, the spectral sensitivity of the camera, etc. It is not possible to find out the exact attenuation coefficient with our limited knowledge and the continuously changing nature of the underwater environment. Recent work [6, 4] has shown theoretically and experimentally that the attenuation coefficients can be modeled as an exponential function, that can be written as,

$$\beta_d(c, d) = j e^{kd} + l e^{md} \quad (4.15)$$

Where j , k , l , and m are the constants that are different for all three color channels and d is the estimated distance between the camera and the underwater scene.

Coefficient Estimation Method

In order to estimate the attenuation coefficient $\beta_d(c, d)$, we first need to estimate the constants in eq 4.15. We do so in a calibration step and use known colors in the image to estimate the coefficients. We put a red, green, and blue color patch in the water and image at different distances. Note that we have a grayscale camera therefore for a given color patch (say red) we can assume that other color channels (say green and

blue) become zero and eq. 4.14 reduces to,

$$I_{gray} = a_c(D_c e^{-\beta_d(c,d)d}) \quad (4.16)$$

Where c is either r (red), g (green), or b (blue) and D_c is the actual color (pixel value) of the patch. We substitute eq 4.15 in eq 4.16,

$$I_{gray} = a_c(D_c e^{-(je^{(kd)}+le^{(md)})d}) \quad (4.17)$$

Given observed dehazed grayscale images i.e., I_{gray}^{observ} vector and estimated distances i.e., a d vector, we formulate an optimization problem as follows

$$\begin{aligned} \min_{D_c, j, k, l, m} \quad & ||I_{gray} - I_{gray}^{observ}|| \\ \text{s/t} \quad & I_{gray}^{observ}(d(0)) \leq D_c \leq 255 \\ & j, k \leq 0 \\ & l, m \geq 0. \end{aligned}$$

Where $I_{gray}^{observ}(d(0))$ is the observed pixel intensity at first distance and it is used as the lower bound on D_c . We run a global search on the values of D_c , j , k , l , and m to estimate the decay coefficients for all three colors separately. Once the constants are estimated, we plug them in eq 4.15 to find the values of the attenuation coefficients.

4.4.4 Color Estimation

Now that we have estimated the distance, removed the veiling effect from the images, and estimated the decay coefficients, the next step is to estimate the colors of the object captured underwater. Recall that in order to reconstruct the colors in a grayscale image, we need to estimate the values of D_r, D_g , and D_b in the eq 4.14. In principle, we have a decay profile² of all the colors in the underwater scene since we have captured images at different distances. We use these decay profiles to estimate

²Decay profile refers to the decay in the pixel values of certain parts of the underwater scene with increasing distance.

the contribution of red, green, and blue in different parts of the object, essentially estimating the RGB channels of the corresponding color image.

Similar to sec. 4.4.3, in this section we formulate an optimization problem for estimating the colors in eq. 4.14. Mathematically, the optimization problem is written as,

$$\begin{aligned} \min_{D_r, D_g, D_b} \quad & \|I_{gray} - I_{gray}^{obser}\| \\ \text{s/t} \quad & 0 \leq D_r, D_g, D_b \leq 255 \end{aligned}$$

We run a global search on this optimization problem in order to estimate the colors in different parts of the object. We then use the estimated D_r, D_g , and D_b to generate red, green, and blue channels of the corresponding color image.

4.5 Implementation and Evaluation

For initial testing of the idea, we used a GoPro camera, suspended and moved with the help of an 80/20 rode structure in a tank. We used a color grid shown in fig. 4-6 as a test object to capture color images. After capturing the images we convert them into grayscale and then use MATLAB for all the post-processing of the images. We run all the tests in a 4x4-meter tank at MIT SeaGrant.

4.5.1 Baselines

We evaluated the qualitative results of our proposed method against state-of-the-art algorithms. Our proposed method essentially colorizes and color corrects the grayscale underwater images in one step to realize ultra-low power underwater color imaging; however past approaches have never explored the possibility of doing so. There is no state-of-the-art color reconstruction (colorization and color correction) method available for underwater grayscale images; however, there is a wide range of separate colorization and color correction methods. We use state-of-the-art colorization and color correction methods in cascade in order to perform color reconstruction

of grayscale underwater images and use them as a baseline. The state-of-the-art methods used are as below:

1. **Colorization:** Colorization is the process of coloring grayscale images. There is no specific underwater image colorization method available therefore we use a general state-of-the-art colorization method to compare the performance of our system. Colorful Image Colorization [81] is a CNN-based colorization method trained on the ImageNet dataset. The results of this work show the improvement in the colorization process against the past methods.
2. **Color restoration:** Color restoration corresponds to restoring the actual colors of an underwater scene. UWCNN [8] is a machine learning-based color restoration method for underwater images. The results show that this model is capable of restoring the colors of colored underwater images in 10 different water types.

4.6 Results

Since this is a testing of the initial idea, we present the theoretical reduction in the power consumption and qualitative results for images and compare them with state-of-the-art systems.

4.6.1 Power Consumption

In this section, we compare the power consumption of our proposed underwater color imaging method against the state-of-the-art battery-free underwater color imaging [3]. Fig. 4-5 shows the power consumption along the y-axis and time along the x-axis; the blue and purple curves represent the power consumption of the SOTA battery-free underwater color camera and WaveColor respectively across time. The spikes in the curve correspond to the time when the camera actively captures an image, whereas the low power ($\sim 20\mu W$) regions correspond to the time when the camera backscatters the image. We make the following remarks

1. The blue curve shows spikes in the power consumption of the system. These spikes correspond to the powering up of LEDs (red, blue, and green) and the CMOS image sensor. Specifically, on average, the LEDs (say red) consume $\sim 8mW$ of power whereas the CMOS sensor itself consumes $\sim 1.1mW$ of power and therefore, the power consumption of the camera is mainly dominated by the power consumption of the LEDs
2. On the other hand, although the power consumption curve of WaveColor (purple curve) shows similar spikes in power consumption when capturing the image, these spikes are caused by the powering of the camera alone, which is 10x less than the power consumption of LEDs.

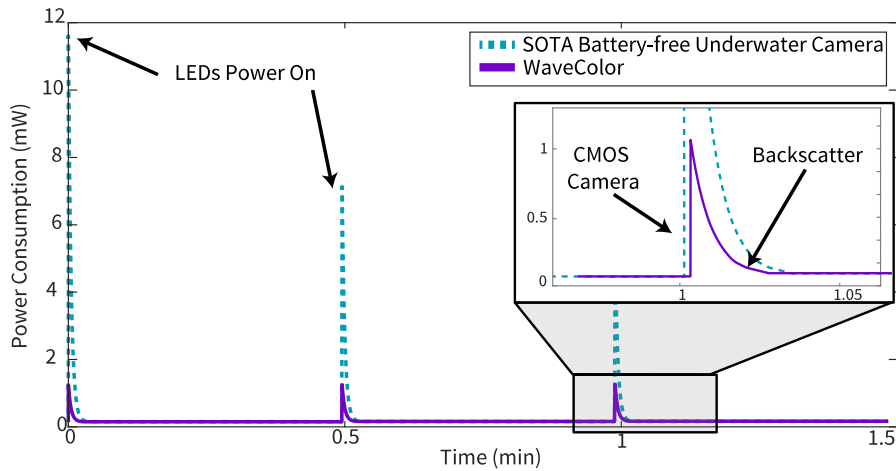


Figure 4-5: **Power Consumption.** The figure shows the comparison between the power consumption of the state-of-the-art battery-free underwater camera (blue curve) [3] and WaveColor (purple curve). The power-limited phase of these cameras corresponds to the time when the camera is actively imaging (the spikes in the curve). The flat (few microwatts) region corresponds to the time in which the camera backscatters the images.

4.6.2 Reconstruction Results

Qualitative Performance

In this section we compare the performance of our proposed method against two methods a) SOTA colorization network only and b) colorization network combined

with SOTA underwater color restoration network UWCNN. Fig. 4-6 shows the qualitative results of the three methods as mentioned earlier (Colorful Image Colorization Method, Colorful Image Colorization Method with UWCNN, and WaveColor). We make the following remarks:

1. It can be seen that the state-of-the-art colorization method is unable to correctly recover the colors of the image. As a result, UWCNN is also unable to restore the true colors of the image. This is because the joint problem of colorization and color correction cannot be simply solved in two steps.
2. It can be seen in the table that WaveColor is able to construct all the colors in the grid. It is to be noted that the results shown in fig 4-6 are preliminary results. Further improvement and methods like user-guided colorization can make the coloring of the images smoother.

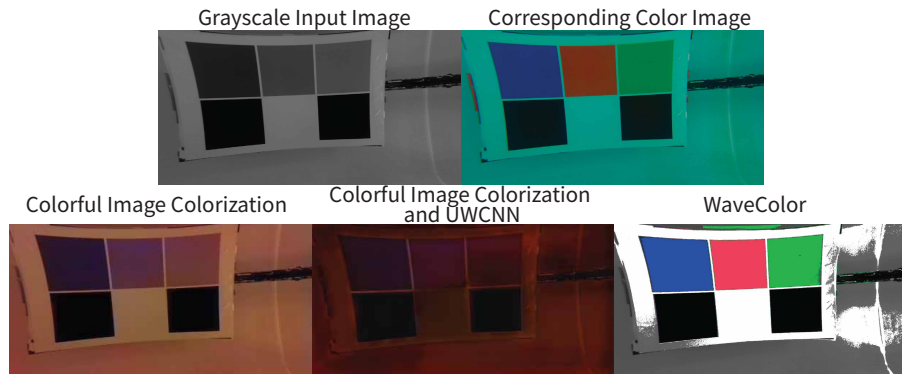


Figure 4-6: **Qualitative Analysis.** The figure shows the qualitative results of the proposed method against the state-of-the-art colorization, and colorization and color restoration method. The first row shows the input grayscale image and the corresponding captured image. The second row shows the results of the three methods (Colorful Image Colorization [81], Colorful Image Colorization followed by UWCNN [8], and WaveColor)

Quantitative Performance

The table 4.1 shows the mean CIEDE2000 (defined in Sec. 2.4.3) of the baselines and our proposed method (WaveColor) on three sets of experiments run with the same tag (shown in fig. 4-6) in different locations and orientations in the tank. CIEDE2000 is

a measure of how different the colors of the image are from another (original) image, which means the smaller the value of CIEDE2000 the better the method is. It can be seen from the table that our method outperforms the baselines by $\sim 85\%$ and $\sim 89\%$ by reducing the CIEDE200 from 84.5 and 109.1 to 11.9 respectively. While these results are preliminary, they demonstrate the promise of our method for color reconstruction.

Method	CIEDE2000
Colorful Image Colorization [81]	84.5
Colorful Image Colorization and UWCNN [8]	109.1
WaveColor	11.9

Table 4.1: **Quantitative Analysis.** The table compares the color difference (CIEDE2000) of the state-of-the-art colorization method, state-of-the-art colorization followed by color correction method and WaveColor

4.7 Conclusion

WaveColor is the first underwater imaging method that is capable of performing ultra-low-power underwater color imaging without using any power-consuming RGB image sensors or LEDs. This method is not only capable of performing underwater color imaging at the lowest power possible but it also captures the true colors of the underwater scene without any color degradation and veiling effect. The proposed method uses undesirable color degradation in underwater images to its benefit thereby revolutionizing the field of underwater image processing.

Chapter 5

Conclusion

This thesis investigates some of the major challenges associated with efficient and ultra-low-power underwater imaging and proposes potential solutions to address them. To this end, we conducted a survey of the challenges faced by ultra-low-power underwater imaging. Subsequently, we presented three solutions to address some of the key challenges, including denoising and desnowing underwater images, enabling ultra-low-power underwater edge inference, and facilitating ultra-low-power underwater color imaging. Specifically, (in Chapter 2) our proposed desnowing method shows $\sim 2dB$ improvement in PSNR and $\sim 17x$ reduction in the required memory. Similarly, (in Chapter 3) this thesis demonstrated edge inference on an ultra-low-power underwater imaging system for the first time that is in both the power and memory budget of the system. Lastly, (in Chapter 4) we presented an ultra-low-power underwater color imaging system, WaveColor, that is capable of reducing the power consumption of the state-of-the-art ultra-low-power color underwater imaging system by $\sim 10x$ and produces colored underwater images with an average color difference of 0.025. Importantly, these solutions are capable of operating within the constraints of ultra-low-power underwater imaging systems.

This thesis paves the way for future research on end-to-end ultra-low-power underwater imaging systems by highlighting the open problems in ultra-low-power underwater imaging and providing avenues for addressing them. Integrating the proposed approaches into an end-to-end system would enable realizing scalable and long-term

underwater imaging. Such imaging would lead to new methods to support long-term underwater observations with applications to monitoring marine ecosystems for applications spanning oceanography, marine biology, and environmental conservation.

Bibliography

- [1] Omniml.
- [2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018.
- [3] Sayed Saad Afzal, Waleed Akbar, Osvy Rodriguez, Mario Doumet, Unsoo Ha, Reza Ghaffarivardavagh, and Fadel Adib. Battery-free wireless imaging of underwater environments. *Nature communications*, 13(1):1–9, 2022.
- [4] Derya Akkaynak and Tali Treibitz. A revised underwater image formation model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6723–6732, 2018.
- [5] Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1682–1691, 2019.
- [6] Derya Akkaynak and Tali Treibitz. Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1682–1691, 2019.
- [7] 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.
- [8] Saeed Anwar, Chongyi Li, and Fatih Porikli. Deep underwater image enhancement. *arXiv preprint arXiv:1807.03528*, 2018.
- [9] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.
- [10] Dana Berman, Deborah Levy, Shai Avidan, and Tali Treibitz. Underwater single image color restoration using haze-lines and a new quantitative dataset. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2822–2837, 2020.

- [11] Vincent Billaut, Matthieu de Rochemonteix, and Marc Thibault. Colorunet: A convolutional classification approach to colorization. *arXiv preprint arXiv:1811.03120*, 2018.
- [12] Henryk Blasinski and Joyce Farrell. A three parameter underwater image formation model. *Electronic Imaging*, 2016(18):1–8, 2016.
- [13] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.
- [14] Wei-Ting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I Chen, Jian-Jiun Ding, Sy-Yen Kuo, et al. All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4196–4205, 2021.
- [15] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE international conference on computer vision*, pages 415–423, 2015.
- [16] John Y Chiang and Ying-Ching Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE transactions on image processing*, 21(4):1756–1769, 2011.
- [17] Corrado Costa, Michele Scardi, Valerio Vitalini, and Stefano Cataudella. A dual camera system for counting and sizing northern bluefin tuna (*thunnus thynnus*; linnaeus, 1758) stock, during transfer to aquaculture cages, with a semi automatic artificial neural network tool. *Aquaculture*, 291(3-4):161–167, 2009.
- [18] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015.
- [19] Yide Di, Xiaoke Zhu, Xin Jin, Qiwei Dou, Wei Zhou, and Qing Duan. Colorunet++: A resolution for colorization of grayscale images using improved unet++. *Multimedia Tools and Applications*, pages 1–20, 2021.
- [20] Moises Diaz-Cabrera, Pietro Cerri, and Paolo Medici. Robust real-time traffic light detection and distance estimation using a single camera. *Expert Systems with Applications*, 42(8):3911–3923, 2015.
- [21] Akshay Dudhane, Praful Hambarde, Prashant Patil, and Subrahmanyam Murala. Deep underwater image restoration and beyond. *IEEE Signal Processing Letters*, 27:675–679, 2020.
- [22] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

- [23] Simon Emberton, Lars Chittka, and Andrea Cavallaro. Hierarchical rank-based veiling light estimation for underwater dehazing. 2015.
- [24] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019.
- [25] Reza Ghaffarivardavagh, Sayed Saad Afzal, Osvy Rodriguez, and Fadel Adib. Ultra-wideband underwater backscatter via piezoelectric metamaterials. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 722–734, 2020.
- [26] S Han, H Mao, and WJ Dally. Deep compression: compressing deep neural network with pruning, trained quantization and huffman coding 4th int. conf. on learning representations (iclr) 2016, conf. *Track Proc*, 2016.
- [27] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.
- [28] Zhu Han, Danfeng Hong, Lianru Gao, Bing Zhang, Min Huang, and Jocelyn Chanussot. Autonas: Automatic neural architecture search for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2022.
- [29] Kevin Peter Hand and Christopher R German. Exploring ocean worlds on earth and beyond. *Nature Geoscience*, 11(1):2–4, 2018.
- [30] Kangjian He, Ruxin Wang, Dapeng Tao, Jun Cheng, and Weifeng Liu. Color transfer pulse-coupled neural networks for underwater robotic visual systems. *IEEE Access*, 6:32850–32860, 2018.
- [31] Inc. Himax Technologies. Hm01b0 ultralow power cis.
- [32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [33] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016.
- [34] Avnet Inc. onsemi arx3a0 cmos image sensor ultra fast rolling shutter, 2023.
- [35] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. Colorization by example. *Rendering techniques*, 29:201–210, 2005.

- [36] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.
- [37] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.
- [38] Jules S Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, 1990.
- [39] Junsu Jang and Fadel Adib. Underwater backscatter networking. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 187–199, 2019.
- [40] Muwei Jian, Xiangyu Liu, Hanjiang Luo, Xiangwei Lu, Hui Yu, and Junyu Dong. Underwater image processing and analysis: A review. *Signal Processing: Image Communication*, 91:116088, 2021.
- [41] Apoorva Joglekar, Devika Joshi, Richa Khemani, Smita Nair, and Shashikant Sahare. Depth estimation using monocular camera. *International journal of computer science and information technologies*, 2(4):1758–1763, 2011.
- [42] Kakani Katija, Giancarlo Troni, Joost Daniels, Kelly Lance, Rob E Sherlock, Alana D Sherman, and Bruce H Robison. Revealing enigmatic mucus structures in the deep sea using deepdiv. *Nature*, 583(7814):78–82, 2020.
- [43] Mohammad Omar Khursheed, Christin Jose, Rajath Kumar, Gengshen Fu, Brian Kulis, and Santosh Kumar Cheekatmalla. Tiny-crnn: Streaming wakeword detection in a low footprint setting. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 541–547, 2021.
- [44] Kenichi Kumatani, Sankaran Panchapagesan, Minhua Wu, Minjae Kim, Nikko Strom, Gautam Tiwari, and Arindam Mandai. Direct modeling of raw audio with dnns for wake word detection. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 252–257, 2017.
- [45] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694, 2004.
- [46] Chongyi Li, Saeed Anwar, Junhui Hou, Runmin Cong, Chunle Guo, and Wenqi Ren. Underwater image enhancement via medium transmission-guided multi-color space embedding. *IEEE Transactions on Image Processing*, 30:4985–5000, 2021.
- [47] Zheng Liang, Yafei Wang, Xueyan Ding, Zetian Mi, and Xianping Fu. Single underwater image enhancement by attenuation map guided color correction and detail preserved dehazing. *Neurocomputing*, 425:160–172, 2021.

- [48] Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, and Song Han. Mcunetv2: Memory-efficient patch-based inference for tiny deep learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [49] Ji Lin, Wei-Ming Chen, Han Cai, Chuang Gan, and Song Han. Memory-efficient patch-based inference for tiny deep learning. *Advances in Neural Information Processing Systems*, 34:2346–2358, 2021.
- [50] Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.
- [51] Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.
- [52] Ji Lin, Ligeng Zhu, Wei-Ming Chen, Wei-Chen Wang, Chuang Gan, and Song Han. On-device training under 256kb memory. *arXiv preprint arXiv:2206.15472*, 2022.
- [53] Peng Liu, Guoyu Wang, Hao Qi, Chufeng Zhang, Haiyong Zheng, and Zhibin Yu. Underwater image enhancement with a deep residual framework. *IEEE Access*, 7:94614–94629, 2019.
- [54] Xiaopei Liu, Liang Wan, Yingge Qu, Tien-Tsin Wong, Stephen Lin, Chi-Sing Leung, and Pheng-Ann Heng. Intrinsic colorization. In *ACM SIGGRAPH Asia 2008 papers*, pages 1–9. 2008.
- [55] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13365–13374, 2021.
- [56] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018.
- [57] Bettina Meyer, Ulrich Freier, Volker Grimm, Jürgen Groeneveld, Brian PV Hunt, Sven Kerwath, Rob King, Christine Klaas, Evgeny Pakhomov, Klaus M Meiners, et al. The winter pack-ice zone provides a sheltered but food-poor habitat for larval antarctic krill. *Nature ecology & evolution*, 1(12):1853–1861, 2017.
- [58] Andrew D Mullen, Tali Treibitz, Paul LD Roberts, Emily LA Kelly, Rael Horwitz, Jennifer E Smith, and Jules S Jaffe. Underwater microscopy for in situ studies of benthic ecosystems. *Nature Communications*, 7(1):1–9, 2016.
- [59] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020.

- [60] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334, 2019.
- [61] Ivan Nagelkerken, Bayden D Russell, Bronwyn M Gillanders, and Sean D Connell. Ocean acidification alters fish populations indirectly through habitat modification. *Nature Climate Change*, 6(1):89–93, 2016.
- [62] Nicholas L Payne, Gil Iosilevskii, Adam Barnett, Chris Fischer, Rachel T Graham, Adrian C Gleiss, and Yuuki Y Watanabe. Great hammerhead sharks swim on their side to reduce transport costs. *Nature communications*, 7(1):1–5, 2016.
- [63] Jack Rademacher, Waleed Akbar, Sayed Saad Afzal, Osvy Rodriguez, Nazish Naeem, Purui Wang, Mario Doumet, Unsoo Ha, Reza Ghaffarivardavagh, and Fadel Adib. Enabling battery-free wireless underwater imaging. In *Proceedings of the 16th International Conference on Underwater Networks & Systems*, pages 1–2, 2022.
- [64] Tara Sainath and Carolina Parada. Convolutional neural networks for small-footprint keyword spotting. 2015.
- [65] Alzayat Saleh, Issam H Laradji, Dmitry A Konovalov, Michael Bradley, David Vazquez, and Marcus Sheaves. A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Scientific Reports*, 10(1):1–10, 2020.
- [66] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [67] Yuya Sato, Takumi Ueda, and Yuichi Tanaka. Marine snow removal benchmarking dataset. *arXiv preprint arXiv:2103.14249*, 2021.
- [68] Ivan W Selesnick, Richard G Baraniuk, and Nick C Kingsbury. The dual-tree complex wavelet transform. *IEEE signal processing magazine*, 22(6):123–151, 2005.
- [69] Gaurav Sharma, Wencheng Wu, and Edul N Dalal. The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, 30(1):21–30, 2005.
- [70] Katherine A Skinner and Matthew Johnson-Roberson. Underwater image dehazing with a light field camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 62–69, 2017.

- [71] Yifan Song, David Nakath, Mengkun She, Furkan Elibol, and Kevin Köser. Deep sea robotic imaging simulator. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part II*, pages 375–389. Springer, 2021.
- [72] STMicroelectronics. Stm32 ultra low power mcus.
- [73] STMicroelectronics. Stm32l476rg ultra-low-power with fpu arm cortex-m4 mcu 80 mhz with 1 mbyte of flash memory, lcd, usb otg, dfsdm.
- [74] STMicroelectronics. X-cube-ai ai expansion pack for stm32cubemx.
- [75] Yosuke Ueki and Masaaki Ikehara. Underwater image enhancement with multi-scale residual attention network. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2021.
- [76] Yiming Wang, Hang Lv, Daniel Povey, Lei Xie, and Sanjeev Khudanpur. Wake word detection with streaming transformers. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5864–5868, 2021.
- [77] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. Transferring color to greyscale images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 277–280, 2002.
- [78] Tidal X. Tidal protecting the ocean while feeding humanity sustainably. <https://x.company/projects/tidal/>, 2021.
- [79] Atsushi Yamashita, Megumi Fujii, and Toru Kaneko. Color registration of underwater images for underwater sensing with consideration of light attenuation. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 4570–4575. IEEE, 2007.
- [80] Ivana Žeger, Sonja Grgic, Josip Vuković, and Gordan Šišul. Grayscale image colorization methods: Overview and evaluation. *IEEE Access*, 9:113326–113346, 2021.
- [81] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.
- [82] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.
- [83] Shu Zhang, Ting Wang, Junyu Dong, and Hui Yu. Underwater image enhancement via extended multi-scale retinex. *Neurocomputing*, 245:1–9, 2017.

- [84] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128*, 2017.
- [85] Lina Zhou, Yin Xiao, and Wen Chen. Imaging through turbid media with vague concentrations based on cosine similarity and convolutional neural network. *IEEE Photonics Journal*, 11(4):1–15, 2019.