Efficient Ultra Low Power Underwater Imaging

Nazish Naeem
MIT
nazishn@mit.edu

Tara Boroushaki
MIT
tarab@mit.edu

Weitung Chen
MIT
weitung@mit.edu

Abstract

Imaging underwater environments can be crucial in advancing our understanding of marine organisms, global climate change, marine geology, aquaculture farms, particulate organic carbon flow, and maritime archeology. Despite all the advances in underwater imaging, researchers estimate that more than 95% of the ocean has never been observed because traditional underwater imaging platforms require active power sources which are unavailable in most underwater environments. 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, paving the way for scalable and long-term in-situ observations of the underwater world. However, due to the low bandwidth of the underwater acoustic channel, capturing and communicating an image takes several hours and makes the overall system energy inefficient.

Building on this work, we propose to develop an efficient low-power underwater imaging system that leverages ultra-low-power edge inference to communicate only useful and necessary information. Specifically, we have developed a fish visual wake word (fishVWW) for battery-free underwater cameras that can trigger the response of the camera to only capture, compress and transmit the image when the camera sees the fish. We have also explored the possibility of removing underwater artifacts from the image before communicating it back to the receiver in order to remove useless information from the transmitting data. We demonstrate the working of the fishVWW model on a low-power microcontroller with very limited memory available.

1 Introduction

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 [27, 28, 17]. Underwater imaging optimizes aquaculture food production, the world's fastest-growing food sector, by detecting diseases such as sea lice and regulating feeding patterns [37]. Underwater imaging also enables the discovery of new marine species and helps us understand the impact of human activities on the marine ecosystem [27, 18].

Despite advances in underwater imaging, more than 95% of the ocean has never been explored or observed. This is because the existing methods for continuous underwater imaging require tethering or bulky batteries for power and communication, limiting their lifetime and scale of operation. As a result, we cannot perform in-situ large-scale underwater imaging. Recent work on battery-free underwater imaging has shown that it is possible to sense the underwater world at scale using battery-free wireless underwater cameras. The camera powers up from the underwater sound, captures color

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

images using an ultra-low-power imaging sensor and low-power LEDs, and communicates images using a new technology called underwater backscatter [13]. The overall design consumes 100,000 times less power than existing underwater wireless cameras, allowing it to operate entirely based on harvested energy.

While the aforementioned advancement is encouraging, these battery-free cameras take a considerable amount of time to communicate an underwater image because of the limited bandwidth of the underwater channel, and they capture low-quality images. Specifically, underwater wireless acoustic channels are inherently bandlimited and can only sustain a few kilobits per second of data rates [16]. As a result, the proposed battery-free camera requires a couple of hours to transmit one color image [3]. This limits the ability of these cameras to monitor moving objects such as fish. Additionally, underwater images suffer from artifacts that limit the usefulness of these images. Specifically, a varied range of turbidity and suspended particles (also known as marine snow) in underwater environments introduce artifacts in the captured images, making these images low-quality and noisy [15]. Hence, this impacts the usability of these images in vision-based applications such as object recognition, classification, and segmentation.

Keeping the above limitations in mind, we ask the following question: *Can we enable near-real-time and high-quality ultra-low-power underwater imaging?* We design an efficient ultra-low-power near-real-time wireless underwater camera. Specifically, we build a fish visual wake word (fishVWW) model and deploy it on an ultra-low power microcontroller. fishVWW helps the camera to capture, compress and communicate the image when it sees the fish in the scene and stays silent otherwise. We also investigate the possibility of desnowing and denoising the underwater images to get rid of underwater artifacts of the images at the edge before communicating them back to the receiver. We have observed that since the addition of noise and snow adds high-frequency components to the image, the compressed noisy image has a much greater size than a clean compressed image. Thus cleaning the images on the edge not only increases the quality of the images captured by the underwater camera but at the same time, it also decreases the communication time of the camera making it both time and energy efficient.

In this project, we demonstrate

- the possibility to perform machine learning inference on battery-free underwater imaging nodes.
- that our fishVWW can be deployed on an ultra low power microcontroller and is able to detect fish in underwater images and disregard transmission of empty underwater images.
- a demo that show the fishVWW model running on a microcontroller with only 65KB of peak memory
- that our tiny convolution network can remove underwater artifacts from images before transmission. However, it need further improvemenmt in order to be deployed on an ultra low power microcontroller.

2 Related Work

2.1 Underwater Imaging:

Underwater imaging has enormous applications but the power-hungry nature of existing underwater imaging systems makes it difficult to enable underwater imaging at scale. For continuous underwater environment monitoring, the vast majority of existing systems either need to be tethered to ships, underwater drones, or power plants for power and communication [18, 7, 21, 29, 36], or in the absence of such tethering, they need to be attached to big batteries which drain out very quickly. This causes impediments to sustainable and long-term underwater observation. To overcome these challenges, recent work [3] has developed a battery-free wireless underwater imaging, which consumes five orders of magnitude less power than reported underwater wireless imaging systems [10, 38, 34]. This method of battery-free underwater imaging is capable of doing passive imaging and active color imaging using low-power active illuminations. It relies on underwater backscatter [14, 11] for energy harvesting and communication which makes its operation battery-free. While [3] paves the way for sustainable underwater imaging, it suffers from high latency and low image quality. The high latency is introduced by the inherent low bandwidth of the wireless underwater acoustic channel. Similarly, the low image quality owes to the imaging artifacts that are specific to the underwater environments, limiting the usefulness of underwater images in vision-based applications.

2.2 Convolutional Neural networks deployment on Microcontrollers:

Running inference and training of neural network on edge devices has various application in health-care, real-time adjustment for agriculture and traffic control, and navigation for smart cars, mobile robots, and drones. The computation resources on microcontrollers are usually limited to less than several Mbs of SRAM and flash, while other mobile devices have access to Gbs of SRAM and Flash storage. As a results, past work has leveraged neural architecture search in order to find an smaller neural network with comparable accuracy to fit the microcontrollers computation resources constraints[6**?** , 22]. Patch-based inference was also suggested and implemented in [23] to alleviate the high memory usage during inference in the initial layers of CNN. Recent work has also shown the feasibility of training on microcontrollers [25] through Quantization-Aware Scaling of gradient and Sparse Update of weights.

However, non of these past work implemented their system on ultra low power microcontrollers or FPGAs and are all battery powered. Ultra low power microcontrollers such as STM32L476RG have more limitation on computation resources and storage.

2.3 Tiny Wake Word Networks

There are multiple application for low-power Wake Work detection. For instance, devices such as Amazon Echo and Apple Siri keep listening for their wake word (Alexa, and Hey Siri) in order to start recording and sending user's request to the cloud for computation and inference. There is extensive past work for audio wake work detection on an edge device. Majority of these systems are not tuned for deployment on ultra low power devices and are focused on detection of wake word in audio signals on cable powered or battery powered devices.[20, 35]

Recent works, [30, 40, 19], have also explored the design and implementation of audio wake word system on low power microcontrollers. However, their techniques and systems does not meet the requirement of underwater imaging and fish visual wake work systems. In particular, audio signals are 1D signals, while images are 2D signals and require more space for saving and more SRAM for processing.

The closest past work is [23] which performs person detection on low resolution images on STM32F412 microcontroller (with 256 kbs SRAM, 1 Mb Flash). In the contrary, our design is intended to enables fish detection based on higher resolution underwater pictures on STM32L476RG, an ultra low power microcontrollers (with 128 kbs SRAM, 1 Mb Flash).

2.4 Underwater Image Denoising and Desnowing:

The problem of denoising and desnowing images is a well-studied problem for in-air captured images, with many algorithms and systems proposed, including those that can run on the edge [2]. Yet, these models cannot be directly applied to underwater imaging since the underwater imaging has inherent differences from in-air imaging including color selectivity, veiling light, marine snow, and underwater backscatter. Not only that, these artifacts also highly depend on water and lighting conditions, making the underwater imaging problem even more challenging [15]. With the emergence of deep learning technologies, there have been several efforts to denoise and enhance these underwater images [39, 26, 32, 12, 41]. However, none of these past approaches were made to enable denoising and desnowing on an edge device with constrains on energy, storage, and computation power.

Apart from the deep learning-based solutions, researchers have also worked on removing the underwater artifacts using different mathematical and image processing techniques like Laplacian pyramids [5], hierarchical processing [9], foreground and background separation [8], and even using special purpose cameras including RGBD and light field cameras [4, 33]. These methods are computationally expensive and water quality dependant thus they are not deployable on low-power cameras and are not generalizable for all underwater imaging environments.

Given the limitations of previous works, this thesis aims at pushing the limits of current underwater imaging systems by enabling efficient and high-quality ultra-low power underwater imaging via edge processing.

3 Method

3.1 Fish Visual Wake Word (FishVWW)

FishVWW like Visual Wake Word(VWW) is a common microcontroller vision use-case for identifying if a fish is present in an image or not. The aim of this part of the project is to implement a VWW for fish in an ultra-low power wireless underwater camera, that acts as a trigger for the camera to capture, compress and communicate only the useful data back to the receiver. This would essentially make the system more efficient. Designing such a method involves the following steps.

- Train a baseline model
- Search the best architecture for a microcontroller given certain constraints
- Finetune the search network
- Quantization and low-level code generation for the model to run on a microcontroller
- Deployment and testing on a microcontroller

3.1.1 Baseline Model

The first step in designing a network for a microcontroller is to train a huge network that can act as a guide/teacher in search of a better network that follows constraints. We trained a baseline model for fishVWW that follows similar architecture as that of a VWW in mcunet [24]. The backbone of the network consists of 14 mobile inverted residual blocks as shown in figure 1. Since the fully connected layer is not supported by the state-of-the-art low-level C library (TinyEngine), we use a pointwise convolution to replace the fully connected layer in the classification layer. The network is trained using an SGD optimizer and cosine warm-up learning rate scheduler over 400 epochs.

Dataset: The network was trained on the DeepFish dataset [31] with 80-20 split for training and validation.

Preprocessing: The data was preprocessed for training to change the images from RGB to grayscale. It is to be noted that the lowest-power camera available is monochrome therefore we processed the data to grayscale before training in order to make it work for ultra-low-power cameras.

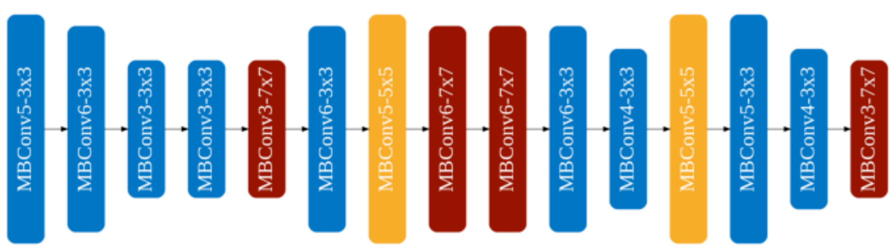

Figure 1: **FishVWW Model:** the figure shows the layerwise visualization of fishVWW model

3.1.2 Neural Architecture Search (NAS)

Once we have a fully trained baseline network, we do a neural architecture search (NAS) on this network to find a network that fits the constraints. Specifically, we use OmniML, a hardware-aware neural architecture search library, to search for a network. We used OmniML with 'autonas' and constraints on parameters (params<=160K). OmniML finds the best network that fits the constraints and gets rid of the extra parameters by means of pruning.

3.1.3 Finetuning

After NAS, we have a model that fits the constraints but has old weights that do not perform well on the test set. The final network needs to be finetuned on the training set in order for it to perform better.

3.1.4 Quantization

After fine-tuning the network, the next step is to convert the float weights/parameters to integers i.e., quantize the neural network. Quantization also affects the performance of the network and needs to be finetuned again. We converted the neural network into a tflite model that does quantization and calibration in the same step i.e., quantization-aware training.

3.1.5 Deployment

After quantization and tflite conversion of the code, we use TinyEngine, a machine learning library for microcontrollers, in order to convert the model into low-level C code that can be deployed on a microcontroller.

3.2 Desnowing Underwater Images

Underwater images in general suffer from underwater artifacts that introduce high-frequency components in the images. The size of these images if compressed is around 2x greater than that of their clean counterparts. If these artifacts are removed at the edge this would not only increase the quality of low-power underwater cameras but also at the same time reduce the amount of data that needs to be transmitted back. This would essentially make the system energy and time efficient.

3.2.1 Baseline Model

Just like the previous network, the first step is to train a baseline model for desnowing the underwater images. We used a fully convolutional network with 6 2D convolutional layers as shown in figure 2 to remove artifacts from these underwater images. Similar to the previous network, this network is trained using an SGD optimizer and cosine warm-up learning rate scheduler over 400 epochs.

Dataset: To train the desnowing network, we used Marine Snow Removal Benchmarking (MSRB) dataset [32]. This dataset is synthesized using a mathematical model of underwater images with marine snow.

Preporcessing: As already mentioned, the ultra-low power underwater cameras are monochrome. Therefore, we transformed the RGB images into grayscale before training the network. The datset was split into training and validation set using 80-20 split.

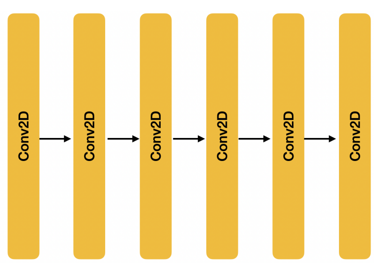

Figure 2: **DesnowNet Model:** the figure shows the layerwise visualization of desnow network

3.2.2 NAS

Similar to section 3.1.2, we used OmniML [1] to search for a smaller network. We put constraints on the parameters and ran 'autonas' in order to search for the candidate network.

3.2.3 Finetuning

After searching for obtaining the best-pruned network, we finetuned the network on the test set to increase the accuracy of the network similar to section 3.1.3

3.2.4 Quantization

Similar to section 3.1.4, after fine-tuning the network, the next step is to convert the float weights/parameters to integers i.e., quantize the neural network. Quantization also affects the performance of the network and needs to be finetuned again. We converted the neural network into a tflite model that does quantization and calibration in the same step i.e., quantization-aware training.

3.2.5 Deployment

The final step is to deploy this network on a microcontroller. For that, we have to generate the model in a language readable by the microcontroller. Similar to section 3.1.5, we used a TinyEngine [24, 23] in order to generate a C code for the microcontroller from tflite model. However, the runtime memory consumption of the network exceeded the maximum available RAM on the microcontroller. We propose to solve this problem by using the following methods:

- We propose to use a NAS, like TinyNAS [24, 23] that is aware of the RAM limits of a microcontroller. We aim to constraint the peakRAM usage in order to find a better network i.e., we want to do a peakRAM aware architecture search
- We also propose to use techniques like patch-based inference [23], in contrast to the classical inference. The patch-based inference has shown a considerable improvement in reducing the peakRAM usage in [23]

4 Results

The evaluation is done on both tasks: fish visual wake word and desnowing net. We will be evaluating the validation accuracy, FLOPS, number of parameters, and projected peak memory usage on the MCU.

4.1 FishVWW

Figure 3 shows the validation accuracy of the baseline mode. We can see that it quickly converges to nearly 100 % of validation accuracy in 100,000 iterations. This baseline model has 22.26M of FLOPS and 334.62K of parameters.

Figure 3: **Validation Accuracy of FishVWW Baseline Model:** the figure shows the validation accuracy of the network against the number of epochs during the training of the baseline model for fishVWW

After we run the Omnimizer and shrink the model size, we fine-tune the model with the same dataset, Figure 4 shows the validation accuracy of the model. It also shows the ability to converge quickly during the finetuning, and also achieve comparable accuracy. However, now we get a big change on the FLOPS and parameters size after running the NAS and pruning. The FLOPS is reported to be 5.41M, and the number of parameters is 146.24K.

With this fine-tuned model, we have converted the model to tflite and generated the MCU code with TinyEngine. The memory usage is shown in Figure 5. The peak memory for the network is 65536 bytes, which is under our system constraint of 128KB. We have run this fish visual wake word on a microcontroller. The demo video can be found here.

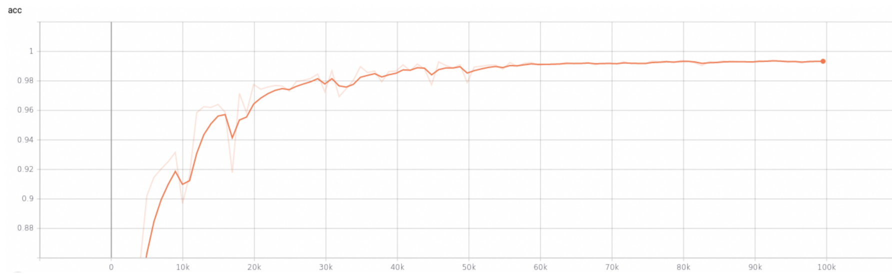

Figure 4: **Validation Accuracy of FishVWW after Omnimizer:** the figure shows the validation accuracy of the network against the number of epochs during finetuning of the fishVWW model after running NAS on it

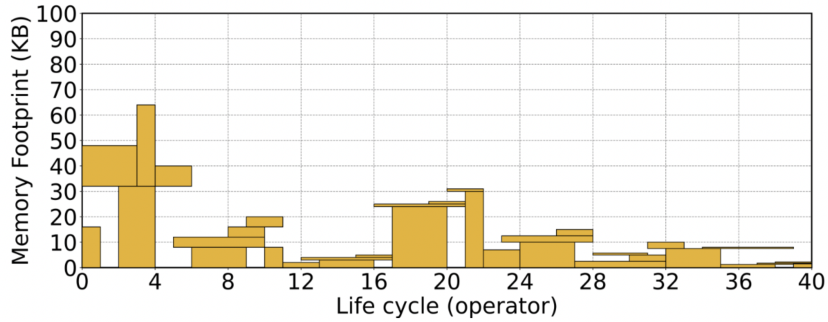

Figure 5: **PeakRAM usage of FishVWW:** the figure shows the peakRAM usage of all the layers of fishVWW model

4.2 Desnowing Underwater Images

For the underwater image desnowing, our metrics of evaluating the model change from prediction accuracy to the root mean square error, which is the difference between the clean image and the image with marine snow. We would like to minimize the root mean square error. Figure 6 shows the root mean square error (RMSE) of the baseline mode throughout the training process. We can see that it reaches an RMSE of 13 at around 300,000 iterations before it jumps and converges toward another local minimum. With this model, it will require 51.77G of FLOPS and 351.07K of parameters.

Figure 6: **Validation RMSE of Desnowing Baseline Model:** the figure shows the validation RMSE of the network against the number of epochs during the training of the baseline model for desnowing

After we run the Omnimizer and shrink the model size, we fine-tune the model with the same dataset, Figure 7 shows the validation RMSE of the model. Now we only need 2.08G in FLOPS and 127.15K in the number of parameters.

Finally, we tried deploying the model on MCU, and Figure 8 shows the memory usage for the model. Unfortunately, the application requires dense output, which causes the activation size to be very large. Therefore, the peak memory usage was evaluated to be 9000KB. Future work on improving this model includes patch-based inference or by modifying the network layers to be less dense.

Figure 7: **Validation RMSE of Desnowing Network after Omnimizer:** the figure shows the validation RMSE of the network against the number of epochs during finetuning of the desnowing model after running NAS on it

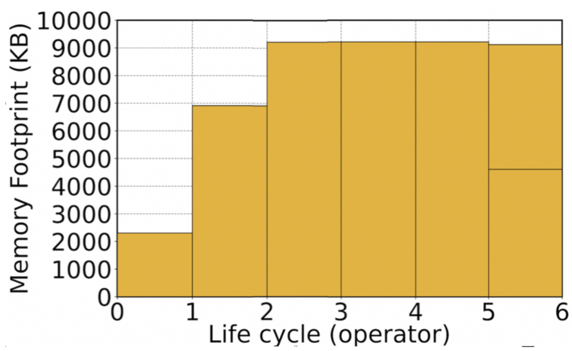

Figure 8: **PeakRAM usage of Desnowing Network:** the figure shows the peakRAM usage of all the layers of desnow model

5 Conclusion

We have explored the possibilities for implementing edge machine learning on battery-free underwater cameras. Our fish visual wake word (fishVWW) edge inference allows the camera to only capture, compress, and transmit images when a fish is detected, saving power and bandwidth. We have also successfully demonstrated the ability to remove underwater artifacts from images before transmission with the tiny convolution network, further reducing the amount of data transmitted. Our work represents an important step towards developing efficient, low-power underwater imaging systems that leverage ultra-low-power edge inference. Our demo shows the viability of our approach by running the fishVWW model on a low-power microcontroller with only 65KB of peak memory.

References

[1] Omniml.

[2] V. Abeykoon, Z. Liu, R. Kettimuthu, G. Fox, and I. Foster. Scientific image restoration anywhere. In *2019 IEEE/ACM 1st Annual Workshop on Large-scale Experiment-in-the-Loop Computing (XLOOP)*, pages 8–13. IEEE, 2019.

[3] S. S. Afzal, W. Akbar, O. Rodriguez, M. Doumet, U. Ha, R. Ghaffarivardavagh, and F. Adib. Battery-free wireless imaging of underwater environments. *Nature communications*, 13(1):1–9, 2022.

[4] D. Akkaynak and T. 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.

[5] C. O. Ancuti, C. Ancuti, C. De Vleeschouwer, and P. Bekaert. Color balance and fusion for underwater image enhancement. *IEEE Transactions on image processing*, 27(1):379–393, 2017.

[6] C. Banbury, C. Zhou, I. Fedorov, R. Matas, U. Thakker, D. Gope, V. Janapa Reddi, M. Mattina, and P. Whatmough. Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers. *Proceedings of Machine Learning and Systems*, 3:517–532, 2021.

[7] Y. Chao. Autonomous underwater vehicles and sensors powered by ocean thermal energy. In *OCEANS 2016-Shanghai*, pages 1–4. IEEE, 2016.

[8] J. Y. Chiang and Y.-C. Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE transactions on image processing*, 21(4):1756–1769, 2011.

[9] S. Emberton, L. Chittka, and A. Cavallaro. Hierarchical rank-based veiling light estimation for underwater dehazing. 2015.

[10] A. Fitzpatrick, A. Singhvi, and A. Arbabian. An airborne sonar system for underwater remote sensing and imaging. *IEEE Access*, 8:189945–189959, 2020.

[11] R. Ghaffarivardavagh, S. S. Afzal, O. Rodriguez, and F. 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.

[12] K. He, R. Wang, D. Tao, J. Cheng, and W. Liu. Color transfer pulse-coupled neural networks for underwater robotic visual systems. *IEEE Access*, 6:32850–32860, 2018.

[13] J. Jang and F. Adib. Underwater backscatter networking. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 187–199. 2019.

[14] J. Jang and F. Adib. Underwater backscatter networking. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 187–199. 2019.

[15] M. Jian, X. Liu, H. Luo, X. Lu, H. Yu, and J. Dong. Underwater image processing and analysis: A review. *Signal Processing: Image Communication*, 91:116088, 2021.

[16] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang. A comprehensive study on the internet of underwater things: applications, challenges, and channel models. *Sensors*, 17(7):1477, 2017.

[17] K. Katija, C. A. Choy, R. E. Sherlock, A. D. Sherman, and B. H. Robison. From the surface to the seafloor: How giant larvaceans transport microplastics into the deep sea. *Science advances*, 3(8):e1700715, 2017.

[18] K. Katija, R. E. Sherlock, A. D. Sherman, and B. H. Robison. New technology reveals the role of giant larvaceans in oceanic carbon cycling. *Science Advances*, 3(5):e1602374, 2017.

[19] M. O. Khursheed, C. Jose, R. Kumar, G. Fu, B. Kulis, and S. K. 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.

[20] K. Kumatani, S. Panchapagesan, M. Wu, M. Kim, N. Strom, G. Tiwari, and A. 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.

[21] K. Lam, R. S. Bradbeer, P. K. Shin, K. K. Ku, and P. Hodgson. Application of a real-time underwater surveillance camera in monitoring of fish assemblages on a shallow coral communities in a marine park. In *OCEANS 2007*, pages 1–7. IEEE, 2007.

[22] E. Liberis, L. Dudziak, and N. D. Lane. nas: Constrained neural architecture search for microcontrollers. In *Proceedings of the 1st Workshop on Machine Learning and Systems*, EuroMLSys '21, page 70–79, New York, NY, USA, 2021. Association for Computing Machinery.

[23] J. Lin, W.-M. Chen, H. Cai, C. Gan, and S. Han. Memory-efficient patch-based inference for tiny deep learning. *Advances in Neural Information Processing Systems*, 34:2346–2358, 2021.

[24] J. Lin, W.-M. Chen, Y. Lin, C. Gan, S. Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in Neural Information Processing Systems*, 33:11711–11722, 2020.

[25] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, C. Gan, and S. Han. On-device training under 256kb memory. *arXiv preprint arXiv:2206.15472*, 2022.

[26] P. Liu, G. Wang, H. Qi, C. Zhang, H. Zheng, and Z. Yu. Underwater image enhancement with a deep residual framework. *IEEE Access*, 7:94614–94629, 2019.

[27] I. Nagelkerken, B. D. Russell, B. M. Gillanders, and S. D. Connell. Ocean acidification alters fish populations indirectly through habitat modification. *Nature Climate Change*, 6(1):89–93, 2016.

[28] N. L. Payne, G. Iosilevskii, A. Barnett, C. Fischer, R. T. Graham, A. C. Gleiss, and Y. Y. Watanabe. Great hammerhead sharks swim on their side to reduce transport costs. *Nature communications*, 7(1):1–5, 2016.

[29] B. H. Robison, K. R. Reisenbichler, and R. E. Sherlock. The coevolution of midwater research and rov technology at mbari. *Oceanography*, 30(4):26–37, 2017.

[30] T. Sainath and C. Parada. Convolutional neural networks for small-footprint keyword spotting. 2015.

[31] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves. A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Scientific Reports*, 10(1):1–10, 2020.

[32] Y. Sato, T. Ueda, and Y. Tanaka. Marine snow removal benchmarking dataset. *arXiv preprint arXiv:2103.14249*, 2021.

[33] K. A. Skinner and M. 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.

[34] M. Tamura, K. Murai, and M. Matsumoto. Design of conductive coupler for underwater wireless power and data transfer. *IEEE Transactions on Microwave Theory and Techniques*, 69(1):1161–1175, 2020.

[35] Y. Wang, H. Lv, D. Povey, L. Xie, and S. 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.

[36] P. H. Wiebe and M. C. Benfield. From the hensen net toward four-dimensional biological oceanography. *Progress in Oceanography*, 56(1):7–136, 2003.

[37] T. X. Tidal protecting the ocean while feeding humanity sustainably. `https://x.company/projects/tidal/`, 2021.

[38] J. Younce, A. Singer, T. Riedl, B. Landry, A. Bean, and T. Arikan. Experimental results with hf underwater acoustic modem for high bandwidth applications. In *2015 49th Asilomar conference on signals, systems and computers*, pages 248–252. IEEE, 2015.

[39] S. Zhang, T. Wang, J. Dong, and H. Yu. Underwater image enhancement via extended multi-scale retinex. *Neurocomputing*, 245:1–9, 2017.

[40] Y. Zhang, N. Suda, L. Lai, and V. Chandra. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128*, 2017.

[41] L. Zhou, Y. Xiao, and W. Chen. Imaging through turbid media with vague concentrations based on cosine similarity and convolutional neural network. *IEEE Photonics Journal*, 11(4):1–15, 2019.