

SonAIr: Real-Time Deep Learning For Underwater Acoustic Spectrum Sensing and Classification

Daniel Uvaydov, Deniz Unal, Kerem Enhos, Emrehan Demirors, Tommaso Melodia
Institute for the Wireless Internet of Things
Northeastern University, Boston, MA, USA
Email: {uvaydov.d, unal.d, enhos.k, e.demirors, melodia}@northeastern.edu

Abstract—Wireless underwater acoustic networking is a key technology with many applications in military and commercial settings. Unlike their radio-frequency (RF) counterpart, underwater acoustic communications are constrained by the limited bandwidth, and impaired by stronger Doppler spread, interference, and frequency selective fading. Furthermore, the underwater channel is completely unregulated. To enable successful underwater spectrum access, spectrum sensing techniques are an important tool to separate ambient interference and undesirable signals from real, desired, transmitted signals. For these reasons, this article introduces SonAIr (Sonar + AI), a real-time system that utilizes Deep Learning (DL) techniques to classify and localize signals in the underwater channel. We collected and curated a real-world underwater acoustic signal dataset spanning 7 days in 2 different locations and multiple transceiver configurations, which will be made publicly available. Additionally, we designed and trained a novel two-stage Convolutional Neural Network (CNN) to identify signal modulations and bandwidths with an Intersection-Over-Union (IOU) as high as 0.89 on days unseen during training. Furthermore, we implemented and validated our classifier in real-time with an underwater testbed, achieving a CNN processing latency of 17 ms.

Index Terms—Deep Learning, Spectrum Sensing, Underwater

I. INTRODUCTION

Underwater acoustic networking is a crucial technology used in both military and civilian sectors, enabling a wide range of applications. These applications include monitoring offshore equipment in wind farms and oil/gas fields [1], tracking global ocean conditions through environmental monitoring [2], improving the efficiency of aquaculture operations through data-driven fish/shellfish ranches [3], and supporting the deployment of autonomous unmanned underwater vehicles for surveillance and monitoring [4]. As the number of these applications grows, so does the demand for shared underwater channel resources within acoustic networks

In contrast to terrestrial communication systems that utilize radio frequency and offer bandwidths ranging from tens to hundreds of MHz per user and tens of GHz in total, underwater acoustic communication channels possess limited and scarce spectrum resources. The available bandwidths range from a few kHz for long-range links of 10 to 100 km, to approximately 10 kHz for medium-range links of 1 to 10 km, and a few tens of kHz for short-range links of 0.1 to 1 km. Only very short-range links of a few hundreds of meters or less may have access to bandwidths exceeding 100 kHz [5].

The lack of regulation exacerbates the scarcity of the underwater acoustic spectrum. Unlike the RF spectrum, underwater

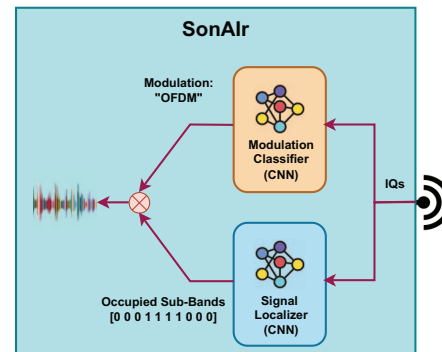


Fig. 1: The SonAIr System. High level overview of real-time underwater spectrum sensing and classification network.

devices do not require a license to function in specific underwater acoustic spectrum bands. There is no strategy for assigning portions of the spectrum and no restrictions on the amount of power that can be transmitted, allowing underwater systems to utilize any part or all of the spectrum whenever they choose with almost unrestricted transmission power levels. However, the spectrum is utilized by other underwater systems used for seismology, profiling, navigation, and marine biology, as well as marine life, which can result in an unpredictable environment for communication links where spectrum resources may experience strong interference both in frequency and time. Despite the overcrowded underwater spectrum, most spectrum resources are still underutilized temporally and spatially. The mobility and low-duty cycles of both man-made and natural acoustic sources can cause spectrum under-utilization temporally, while directional acoustic sources and nonlinear sound propagation can lead to spectrum under-utilization spatially.

Overall, it is evident that the scarce and underutilized spectrum resources, compounded by the lack of regulation and the overcrowded spectrum, pose a significant limitation on the widespread use of underwater communication and networking systems. Spectrum sensing and classification systems have the potential to alleviate the aforementioned spectrum-associated problems and enable the prevalent use of wireless communication and networking underwater. By continuously monitoring the acoustic spectrum and identifying the available frequency bands, such systems can enable underwater devices to utilize the available spectrum more efficiently and avoid interference with other systems that operate in the same frequency bands. Additionally, by automatically sensing and

classifying allocated bandwidths and communication protocols, such systems can distinguish transmissions from interference and noise, minimize the need for handshaking/signaling and feedback overhead between transmitter and receiver pairs, and allow for efficient communication in unregulated, possibly contested environments where multiple acoustic sources coexist.

To address the existing lack of real-time spectrum sensing and classification systems, this paper proposes *SonAIr* (Sonar + AI), a real-time system that leverages Deep Learning (DL) techniques, particularly convolutional neural networks (CNNs), to sense, classify, and localize signals in the underwater channel. Traditional underwater spectrum sensing and classification systems rely on protocol-specific feature extraction techniques, which may be limited in terms of adaptability, generalizability, accuracy, robustness, and real-time capability [6]–[8]. In contrast, *SonAIr* performs sensing and classification on unprocessed in-phase and quadrature (I/Q) samples to classify physical layer (PHY) protocols and localize them in frequency without any prior knowledge, as shown in Fig. 1. It can offer sensing and classification performance with an Intersection-Over-Union (IOU) as high as 0.89 on days not seen during training, with a processing latency of 17 ms.

We summarize our key technical contributions in this paper as follows:

- We have created a real-world dataset¹ of underwater acoustic signals collected over a 7-day period in two different locations and utilizing multiple radio configurations.
- We have designed and trained a two-stage signal classifier based on Convolutional Neural Networks (CNNs) that is capable of accurately localizing signals as narrow as 2.5 kHz and identifying up to 3 physical layer modulations, achieving a mean Intersection-Over-Union of up to 0.89 on days not seen during training.
- We have implemented and validated the effectiveness of our classifier in real-time, with a processing latency of only 17 ms for the CNN component.

The rest of the paper is organized as follows: in Section II we review the state of the art in underwater spectrum sensing, in Section III we introduce the 7 day dataset we have curated containing real underwater acoustic signals, in Section IV we review our DL algorithm for underwater spectrum sensing, in Section V we present our results, and finally in Section VI we draw the main conclusions.

II. RELATED WORK

DL has been widely adopted and researched in wireless communications in recent time for both inference and control purposes, however predominantly in the RF domain [9], [10]. As mentioned before, due to the harsher conditions in the underwater channel [11], DL for underwater communications does not have the widespread adaptation of its RF counterpart. While not a brand new idea in the underwater space, DL is still in its infancy in this domain. Most works applying DL to aid underwater communications are limited by simulated datasets

with assumed channel models [12], [13]. While this helps validate the theoretical feasibility of such implementations it does not translate well to real-world applications. There are works that attempt at minimizing the translational challenge to real world implementation with different techniques. These works generally train their systems with simulated data using novel algorithms, then validate with experimental data. [14] develops a blind signal detector for underwater acoustic signals, essentially distinguishing the signals from noise, and developing a transfer model reducing the reliance on simulated data for online testing. [15]–[17] develop a modulation classifier for underwater acoustic channels using DL techniques and validate in data from real experimental scenarios. While the results of many of these works are promising and are shown to perform with data from real underwater scenarios, they are limited in identification complexity. For example, none of them localize the signals in addition to detecting it or classifying it. This is generally the trend with works first trained on simulated data that translate over to real underwater signals, as the capability for more complex DL tasks is waning without an established real-world dataset.

To the best of our knowledge, no work exists that aims to perform classification on underwater acoustic signals with abilities to identify and localize multiple signals. Furthermore, no real-time testbeds exist that can perform such a task. Finally, there does not exist a dataset that will help enable underwater spectrum sensing to progress and ease the burden of data scarcity. This work aims to bridge all of these gaps by providing data to further the field of underwater modulation recognition and spectrum sensing as well as implementing a state-of-the-art working system.

III. DATASET

To develop and test *SonAIr*, we produced a dataset by recording transmissions with varying modulation and frequency bands using deployed underwater acoustic modems. Existing commercial underwater acoustic modems [18], [19] typically utilize non-coherent techniques such as frequency shift keying (FSK) and chirp spread spectrum (CSS), which provide robust links and have low-complexity design. Whereas research modems [20] make use of multi-carrier coherent techniques such as orthogonal frequency division multiplexing (OFDM), as these can achieve higher rates at a cost of higher complexity. The dataset for *SonAIr* comprises packets utilizing FSK, CSS, and OFDM to encompass the physical layer implementations commonly used in underwater acoustic communication systems.

The parameters of these classes are varied based on the effective bandwidth of the sample. For FSK, symbols hop between $\{-f_{BW}/4, f_{BW}/4\}$, and the rate parameter is adjusted to cover total bandwidth. For CSS, chirp duration is set to 2 ms and sweep frequencies coincide with the total bandwidth. For OFDM, a subcarrier bandwidth of 19.5 Hz is selected, and the number of subcarriers is scaled based on the total bandwidth. To make the spectrum sensing problem more manageable, we have divided the spectrum into equally spaced channels. We further assume that each transmission is aligned with channels

¹<https://github.com/wineslab/sonair-dataset>



Fig. 2: Marina deployment with two underwater acoustic modems for data collection

such that the center frequency of the transmitter is set to one of the channel boundary frequencies while the receiver center frequency is fixed. The transmission bandwidth is an integer multiple of the channel bandwidth, but unintentional leakage to adjacent channels is permissible and additional filtering is not used on the transmitter or receiver.

In this work, we consider 20 channels with 2.5 kHz channel bandwidth over a 50 kHz total bandwidth. A set of raw data files for 5, 10, 15, 20, 25, and 50 kHz bandwidths for each modulation class is generated offline. The bandwidth and transmitter center frequency parameter pair determines the set of occupied channels. The files have randomly generated payloads to avoid any distinct patterns in the recordings and to limit peak to average power ratio (PAPR) for OFDM recordings.

The dataset generation process is controlled by a server that has access to both of the modems. A script running on the server sequentially processes each recording. The main tasks of this script are configuring transmitter and receiver modems, starting the transmitter, and after the transmitter is active, recording for the specified duration of time with the receiver modem. On the modems, GNU Radio is used to play and record the files. The configuration step on the transmitter side consists of loading appropriate files based on modulation and bandwidth, setting the center frequency to occupy given channels, and setting the output level. The receiver modem is always configured with the system center frequency. Recording a dataset takes approximately 20 to 50 minutes including background noise recordings. The dataset consists of binary files of baseband interleaved 32-bit float I/Q samples. The files are labeled with the corresponding class followed by an array of '1' and '0' representing occupied and empty channels respectively. For instance, a recording of CSS-modulated packets of 15kHz bandwidth occupying channels 7-12 is labeled "css_0000000111110000000.dat".

We have collected three sets of data on different days. Six days were recorded at a marina and one set is from an open sea experiment. The typical marina setup consists of two modems attached to the dock with ropes and suspended next to each other as illustrated in Fig. 2. The projector of the transmitter

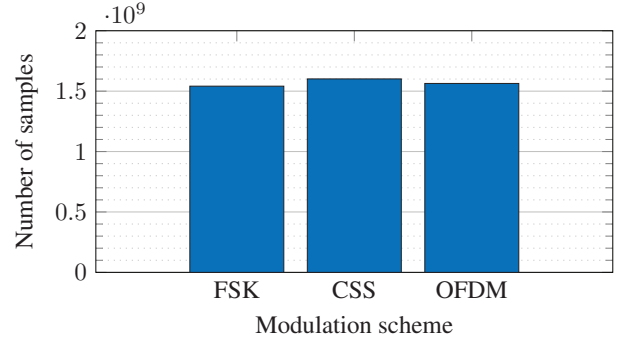


Fig. 3: Total number of I/Q samples per class

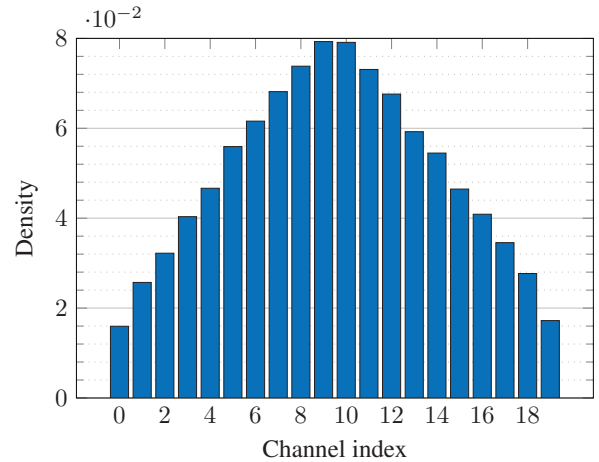


Fig. 4: Distribution of I/Q samples per channel

modem faces up whereas the hydrophone of the receiver modem faces down. The resulting vertical setup has a separation of 5 to 7 meters. This setup resembles a typical topology in which a sink node close to the surface receives information from underwater nodes deployed below.

The recordings at sea were performed near Newport, Rhode Island. The modems were attached to buoys and deployed far from the shore. The initial placement and orientation were similar to the marina setup, but the vertical separation of the modems was approximately 20 meters and their horizontal alignment was varying due to currents.

The total duration of recordings in this dataset is two hours, twelve minutes, and two seconds including empty channel recordings. The number of I/Q samples for each modulation scheme in these recordings is given in Fig. 3. The channel usage was not uniform across all frequency bins due to the generation procedure described earlier, and the resulting distribution is given in Fig. 4.

The center frequency of the receiver is set to 150 kHz which is determined according to the optimum transmit frequency response of the transducer. The resulting channel mapping spans the 125kHz to 175kHz interval. A spectrogram of typical background noise levels from marina recordings is shown in Fig.

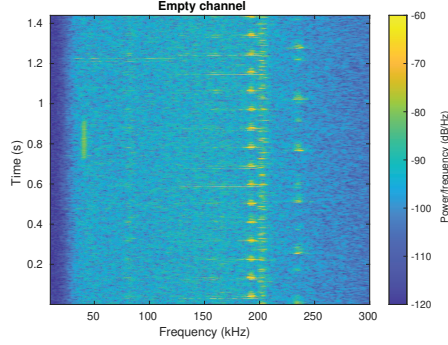


Fig. 5: Spectrogram of empty channel

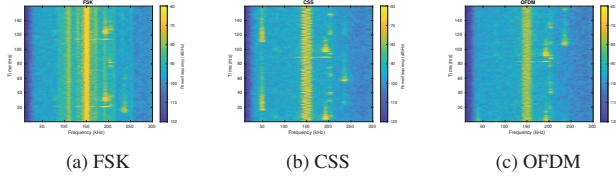


Fig. 6: Spectrogram of different modulation classes with 20 kHz bandwidth

5. Here, transmissions from various acoustic sources around the marina are captured with the most prominent one around 200 kHz. Spectrograms of short sections from recordings of the three classes from the same day are given in Fig. 6. In these plots, various noise components are observed.

IV. DEEP LEARNING AIDED SPECTRUM SENSING

A. System Architecture

The full system architecture of *SonAir* can be seen in Fig. 7. The modems are the software-defined underwater acoustic modem described in [21], and the server is a Dell T440 configured with a Intel Xeon 4208 CPU and 32 GB RAM. The signal received with the transducer is amplified with the receiver hardware and passed to the digital downconverter on the programmable logic section of the modem. The signal is downmixed from center frequency to baseband and decimated to 2.5 Msps. The output is then streamed into a GNURadio flowgraph that is running on the processing system of the modem. In this flowgraph, the received baseband signal is further decimated to 50 ksamples/s. The decimation process filters out the out-of-band noise and reduces the rate of data to be streamed and processed at the following spectrum sensing blocks. The resulting signal at the output of the decimation block precisely occupies the total bandwidth considered for the spectrum sensing blocks which work in parallel and will be discussed in more detail in Section IV-B. Both spectrum sensing blocks take 400 I/Qs at this new sampling rate as input.

Real-time inference for these spectrum sensing blocks is performed with an existing GNU Radio Out of Tree (OOT) module for CNN inference on GNU Radio [22]. This module utilizes ONNX and ONNX runtime to run DL modules converted to

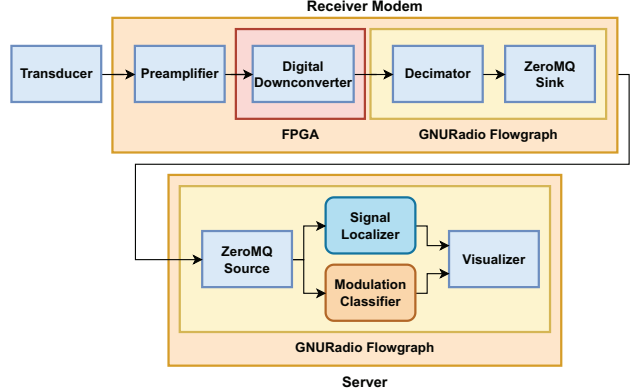


Fig. 7: *SonAir* system architecture

the ONNX format, optimized for hardware acceleration and interoperability. In our flowgraph we have used two of these modules in parallel for the Modulation Classifier and the Signal Localizer models. ONNX model files used to initialize these blocks are produced from the network parameters obtained after training. The networks output a vector of probabilities for each set of 400 I/Q symbols that have been fed as input. The output of the Modulation Classifier is a 1D vector of size 4, each index corresponding to a different class (PHY modulation). This vector is fed into an argmax block that calculates the index of the most probable class. The Signal Localizer output vector, on the other hand, is a 1D vector of size 20 (each index corresponding to a different 2.5kHz sub-band in the observable bandwidth) that is passed through a threshold function that maps $[0, 0.5)$ to 0 and $[0.5, 1]$ to 1. The resulting scalar and vector outputs are multiplied to a compact form of vector of class indices. Fig. 8 shows the joint performance of both classifiers and how the results are combined to create one cohesive label representing the current state of the spectrum.

B. Two Stage Spectrum Classifier

Our primary vehicle for driving underwater spectrum sensing are 1D CNNs, specifically we design two 1D CNNs that operate in parallel. (i) The *Modulation Classifier*, classifies the modulation used for the transmitted signal and (ii) the *Signal Localizer*, finds the occupied frequencies of said signal. Both of these CNNs take the exact same input and have the same structure, but have different outputs and different amount of filters in each CNN layer. From our experiments we received better performance when dividing the tasks this way as opposed to one network that performs both functions. Furthermore dividing the task in this way does not allow classification errors to propagate from one task to another, as the networks are not trained to optimize both at the same time.

Both CNNs take a 2 channel (real and imaginary) input of 400 raw I/Qs sampled at 50kHz in the time domain. "Channel" referring to the channels of a CNN and not the frequency channels in the wireless spectrum. This input size is chosen through training iterations of sizes and is found to be the smallest size

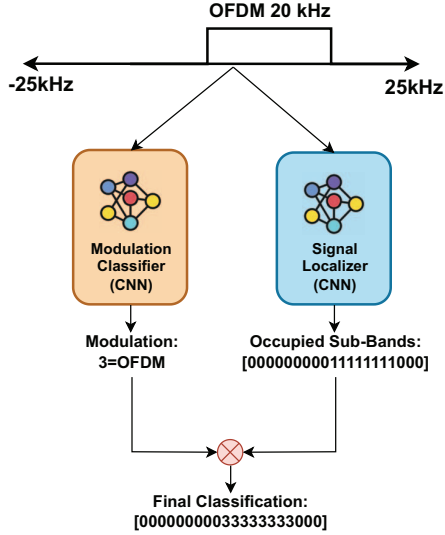


Fig. 8: Joint performance of Modulation Classifier and Signal Localizer on classifying wireless acoustic underwater signals

with the best performance for both networks. Before feeding into the networks the I/Q s are normalized by setting their l_2 norm to 1 or unit power as can be seen in Fig. 9. Although this removes any information the magnitude of the constellation vector may contain (as they are all unitary now), this will ensure the network doesn't rely on signal power or signal-to-noise ratio (SNR) to make its distinctions. Rather, the network must learn from the phase patterns of the constellations in the I/Q plane. The intuition is that the received signal strength will play less of a factor in classification and the network will generalize better in more diverse scenarios or environments. The classifier therefore should be adaptable.

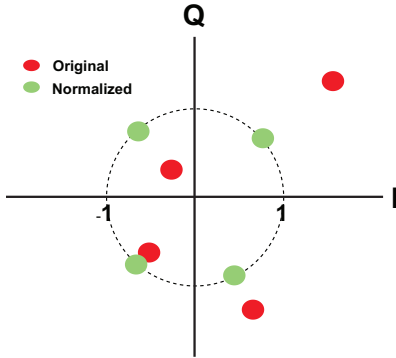


Fig. 9: I/Q constellation change after l_2 normalization

To keep our network lightweight we employ a altered version of a VGG16 [23] network as seen in Fig. 10. The VGG16 network is used widely for a variety of signal and spectrum classification related tasks and has shown great performance. Specifically this network performs very well as a general-

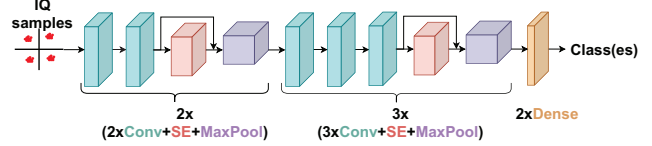


Fig. 10: CNN used for feature extraction for both Modulation Classifier and Signal Localizer. Inputs are the same for both but output varies depending on module.

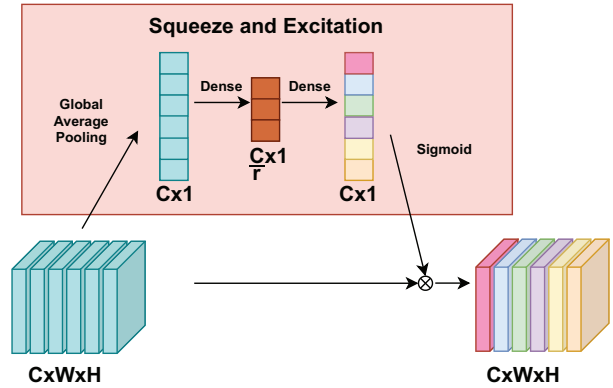


Fig. 11: Squeeze and excitation network that calculates global weights for each CNN channel and then scales each channel by these weights

ized feature extractor. However, we add a channel attention mechanism inside each convolutional block via squeeze-and-excitation (SE) networks [24]. SE networks provide channel attention by multiplying each channel at a given layer by a set of weights based on the global average of each channel and some feature extraction as seen in Fig. 11. Here the global average of each channel is calculated than put through a mini encoder decoder dense network (the "squeeze" and "excitation" aspect) where r controls the squeeze factor. The network is then excited to have as many neurons as there are channels of the initial feature map for scaling. Furthermore due to the Sigmoid activation at the final dense layer of the SE network before scaling, each channel is weighted based on importance or impact on minimizing the loss function. This will allow the model to adaptively choose which channel it deems better for the classification. For our application this allows context awareness where certain aspects of the constellation or I/Q plane can be deemed more pertinent than others. This is especially important as we have removed the magnitude information, so we force the network to focus on other aspects. The SE network is also a very computationally inexpensive way to add channel attention as it is not made up of any layers not already used in traditional CNNs.

Modulation Classifier. The Modulation Classifier's sole purpose is to identify the modulation scheme of the transmitted signal. As discussed in Section III our signals are modulated either using CSS, FSK, or OFDM. Furthermore we add an extra class for noise and interference when no signal is present which we call "Empty". This will allow for the network to account for

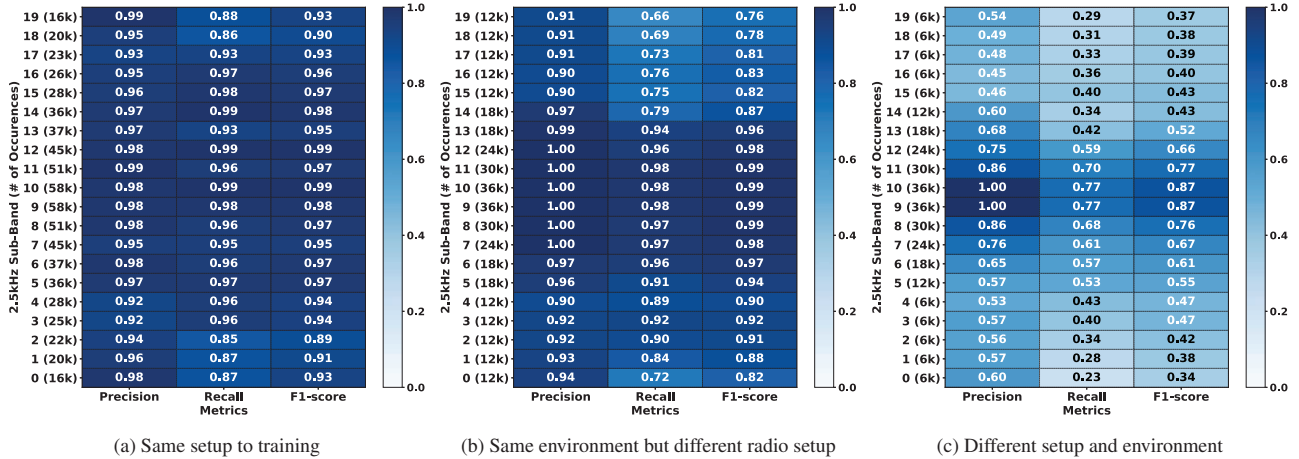


Fig. 12: Classification report of *SonAir* in three different configurations on days unseen during training

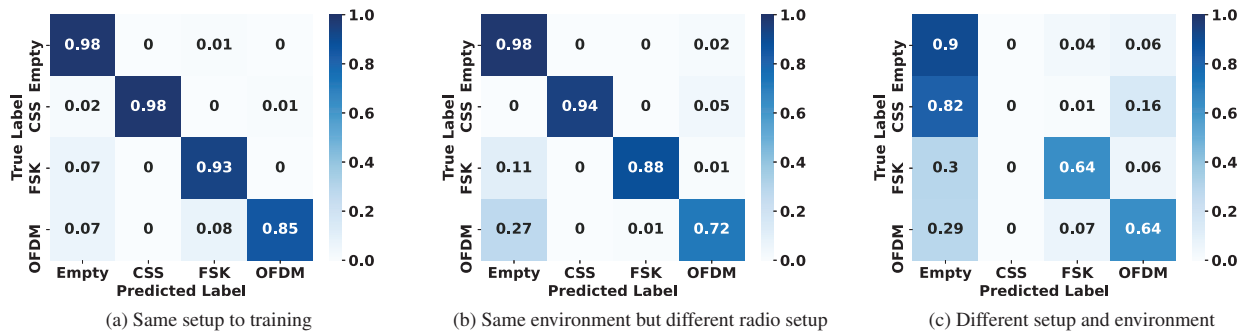


Fig. 13: Confusion matrices of *SonAir* in three different configurations on days unseen during training

any undesirable signals, whether they be other modulations or ambient noise/interference. This network outputs a single integer which is attributed to a modulation as follows: $Empty = 0$; $CSS = 1$; $FSK = 2$; $OFDM = 3$. The network is trained in the traditional single-label classification way with a categorical cross-entropy loss function, as we assume only one modulation can exist within the observable bandwidth. An Adam optimizer is used for the gradient descent.

Signal Localizer. The Signal Localizer as a binary multi-label classifier where each output neuron is attributed with a different sub-band in the observable band. In this case we have a 50kHz observable band and 20 output neurons, so each output neuron dictates whether there is a signal present in 2.5kHz of bandwidth. In a sense the Signal Localizer acts as a form of adaptive energy detector without any kind of a threshold. The benefits of using a DL network over a fixed threshold energy detection algorithm has been shown in the literature before [25]. The network will "fire" any neuron where it detects a signal of interest. Therefore this network outputs a vector of size 20 filled with ones and zeros, ones if there is a signal and zero if there isn't. An Adam optimizer is used for the gradient descent here as well.

V. RESULTS

A. Classification

As mentioned previously we collect data for 7 days, three for training and four for testing. Partitioning data in this way allows us to prioritize testing the generalization capability of our classifiers to different days. The four testing days have different radio configurations and environments. We first report the performance of the Signal Localizer as a binary multi-label (multi-hot encoded) classifier, to show the classification capability on a sub-band level. We then report the joint performance of the Signal Localizer and the Modulation Classifier.

The three plots in Fig. 12 report the precision, recall and F1-score of the Signal Localizer in each 2.5kHz sub-band, showing its ability to identify any signal present for a sub-band irrespective of others. Each plot represents a different environment and physical radio setup on a day different than used for training or "unseen" to the classifier. Specifically Fig. 12a reports the classification performance when the radios are in the exact same setup and environment (testing day 1 and 2), Fig. 12b reports a different setup but in the same environment (testing day 3), and Fig. 12c reports a different setup and environment (testing day 4). We can see the Signal Localizer shows high performance

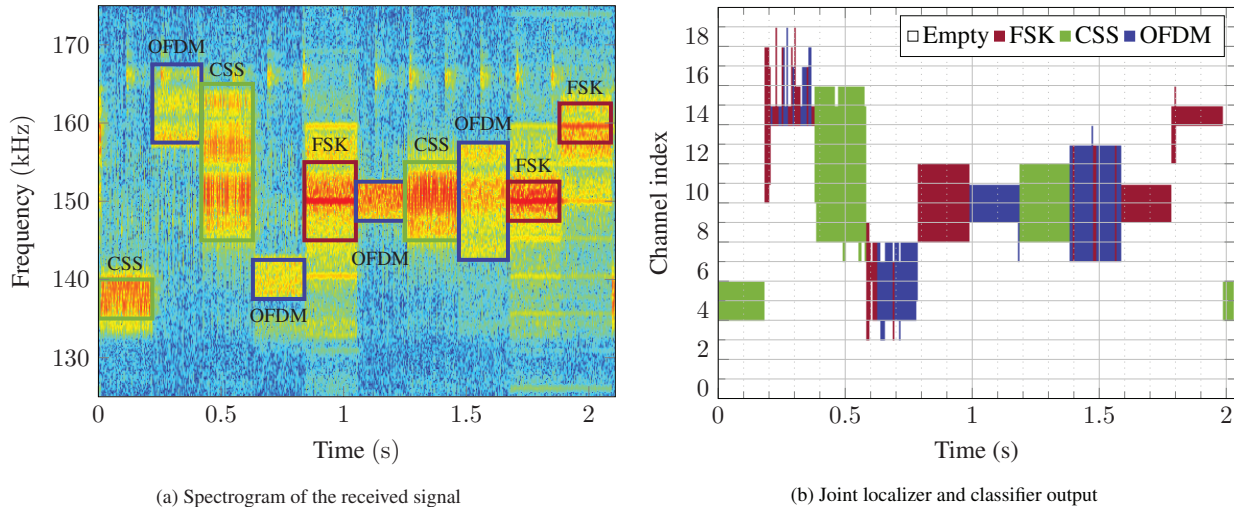


Fig. 14: Real-time inference on GNU Radio

when tested on days in the same environment as seen in Fig 12a and 12b regardless of the radio setup or the day, showing that as long as we are in a relatively similar body of water as used for training, we can attain high generalization regardless of the day or radio locations. We can also see that class imbalance plays less of an effect in these two cases, as seen by fairly consistent performance at the edge bands despite having fewer occurrences or presence in the dataset. Class imbalance plays more of an effect on performance as the environment becomes more foreign to the classifier. This is prevalent in Fig. 12c as the center sub-bands with more occurrences during training and testing perform very well, while the edges suffer.

Fig. 13 shows the confusion matrices of the joint performance of the Modulation Classifier and the Signal Localizer or the full system performance. As before each confusion matrix corresponds to different testing days. When comparing the overall performance across different testing days we see a similar trend as before, where *SonAir* has better performance when tested in similar bodies of water. Looking at the performance across modulations, we can see that OFDM classification becomes difficult as the radio setup is changed within the same environment, where some of the sub-bands are confused for an empty channel. We observe this more for the edge sub-carriers. As the location becomes more foreign however, OFDM performance stops degrading and CSS and FSK becomes harder to classify. This is due to the fact that these signals generally take up less of the spectrum and can be easily confused for noise.

To give a holistic understanding of our two-stage classifier’s performance we report the mean Intersection-Over-Union or IOU in Table I. IOU represents the amount of overlap that the classified signal has with the true signal in frequency as defined by,

$$\text{Mean IOU} = \frac{1}{N} \sum_{n=1}^N \frac{\text{Area Of Intersection}_n}{\text{Area Of Union}_n}. \quad (1)$$

where n is one of the modulation classes and N is the total number of modulation classes. For our purposes this represents the amount of overlap in frequency and is used widely in semantic segmentation and object detection tasks in computer vision. We can see, again, that in the same body of water, we have very high sensing performance and strong degradation in more foreign bodies.

TABLE I: Mean IOU of two stage classifier on different days unseen during training

Day	Mean IOU
Same setup to training	0.89
Same environment but different radio setup	0.83
Different setup and environment	0.40

B. Real-Time Performance

The real-time operation of the *SonAir* system is implemented as described in Section IV-A and demonstrated in Fig. 14. This experiment was performed when modems were deployed at the marina, on a different day than dataset recording. From the transmitter side, a predefined sequence of input signals is transmitted. On the receiver side both the combined output from the DL blocks and the received signal is recorded at the server. In Fig. 14a, the spectrogram of the downsampled input signal is shown with bounding boxes showing the true signal labels. The output of the Visualizer block indicating classification result is given in Fig. 14b. For the Visualizer output, the x-axis indicates time, the y-axis denotes channel indices, and the color axis indicates classifier output for a channel index. Three discrete colors of red, green, and blue can be displayed which correspond to FSK, CSS, and OFDM classes as well as the white background which corresponds to Empty. It is easy to see that the classifier output directly corresponds with the real spectrogram bounding box labels and is able to accurately

localize the signal as well as identify the modulation. The latency incurred by the CNN inference block is 17ms with a standard deviation of 16ms when run on a CPU.

VI. CONCLUSIONS

We have presented *SonAir*, a DL enabled spectrum sensing system that is able to identify and localize underwater acoustic signals with a IOU as high as 0.89. *SonAir* has also been implemented in a real-time testbed and can classify signals with an average latency of 17 ms. Furthermore, *SonAir* shows good generalization performance within a similar body of water to that used for training, even when the radio locations and setup is different. However, in entirely new bodies of water, performance starts to degrade and we hope that future work will address this shortcoming in creating environment-invariant spectrum awareness when using DL. To enable future research, we have curated a 7-day dataset with real underwater acoustic signals that we have used to train, test, and validate *SonAir* and pledge to make this dataset publicly available to help aid future research in underwater spectrum sensing.

REFERENCES

- [1] Forbes, "Drones begin to deliver on their potential for the oil and gas sector," 2018, <https://tinyurl.com/s8yo6ka9>.
- [2] H. G. Kim, G. H. Kim, and H. J. Lee, "A review of the applications of underwater sensor nodes for oceanographic data collection and monitoring," *Sensors*, vol. 10, no. 9, pp. 7958–7975, 2010.
- [3] C. Wang, Z. Li, T. Wang, X. Xu, X. Zhang, and D. Li, "Intelligent fish farm—the future of aquaculture," *Aquaculture International*, vol. 29, pp. 1801–1816, 2021.
- [4] J. González-García, A. Gómez-Espinoza, E. Cuan-Urquizo, L. G. García-Valdovinos, T. Salgado-Jiménez, and J. A. E. Cabello, "Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions," *Applied Sciences*, vol. 10, no. 4, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/4/1256>
- [5] T. Melodia, H. Kulhandjian, L. Kuo, and E. Demirors, "Advances in underwater acoustic networking," in *Mobile Ad Hoc Networking: Cutting Edge Directions*, 2nd ed., S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Inc., Hoboken, NJ: John Wiley and Sons, 2013, pp. 804–852.
- [6] Y. Luo, L. Pu, M. Zuba, Z. Peng, and J.-H. Cui, "Challenges and opportunities of underwater cognitive acoustic networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 198–211, 2014.
- [7] Y. Miao, Y. V. Zakharov, H. Sun, J. Li, and J. Wang, "Underwater acoustic signal classification based on sparse time–frequency representation and deep learning," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 3, pp. 952–962, 2021.
- [8] C. Chin-Hsing, L. Jiann-Der, and L. Ming-Chi, "Classification of underwater signals using wavelet transforms and neural networks," *Mathematical and computer modelling*, vol. 27, no. 2, pp. 47–60, 1998.
- [9] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: A comprehensive survey," *Ad Hoc Networks*, vol. 93, p. 101913, 2019.
- [10] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- [11] R. A. Khalil, N. Saeed, M. I. Babar, and T. Jan, "Toward the internet of underwater things: Recent developments and future challenges," *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 32–37, 2020.
- [12] Y. Zhang, J. Li, Y. Zakharov, X. Li, and J. Li, "Deep learning based underwater acoustic ofdm communications," *Applied Acoustics*, vol. 154, pp. 53–58, 2019.
- [13] Y. Wang, H. Zhang, Z. Sang, L. Xu, C. Cao, and T. A. Gulliver, "Modulation classification of underwater communication with deep learning network," *Computational intelligence and neuroscience*, vol. 2019, 2019.
- [14] Y. Li, B. Wang, G. Shao, S. Shao, and X. Pei, "Blind detection of underwater acoustic communication signals based on deep learning," *IEEE Access*, vol. 8, pp. 204 114–204 131, 2020.
- [15] W. Zhang, X. Yang, C. Leng, J. Wang, and S. Mao, "Modulation recognition of underwater acoustic signals using deep hybrid neural networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 5977–5988, 2022.
- [16] Y. Wang, Y. Jin, H. Zhang, Q. Lu, C. Cao, Z. Sang, and M. Sun, "Underwater communication signal recognition using sequence convolutional network," *IEEE Access*, vol. 9, pp. 46 886–46 899, 2021.
- [17] W. hua Jiang, F. Tong, Y. ze Dong, and G. qiang Zhang, "Modulation recognition of non-cooperation underwater acoustic communication signals using principal component analysis," *Applied Acoustics*, vol. 138, pp. 209–215, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0003682X16304443>
- [18] Teledyne Marine, "Acoustic Modems," Last checked: 2023. [Online]. Available: <http://www.teledynemarine.com/acoustic-modems/>
- [19] EvoLogics, "Underwater Acoustic Modems," Last checked: 2023. [Online]. Available: <https://evologics.de/acoustic-modems>
- [20] S. Mangione, G. E. Galioto, D. Croce, I. Tinnirello, and C. Petrioli, "A channel-aware adaptive modem for underwater acoustic communications," *IEEE Access*, vol. 9, pp. 76 340–76 353, 2021.
- [21] D. Unal, S. Falleni, E. Demirors, K. Enhos, S. Basagni, and T. Melodia, "A software-defined underwater acoustic networking platform for underwater vehicles," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 2531–2536.
- [22] O. Rodriguez and A. Dassatti, "Deep learning inference in gnu radio with onnx," in *Proceedings of the GNU Radio Conference*, vol. 5, no. 1, 2020.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [24] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [25] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "DeepSense: Fast wideband spectrum sensing through real-time in-the-loop deep learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.