ShipNN: On-Device, Ship Identification from Underwater Noise

Kainat Altaf[1], Momin Ali[1], Laura Harms[1,2], Christian Renner[2], Olaf Landsiedel[2,1]
[1]*Kiel University*, Germany
[2]*Hamburg University of Technology*, Germany
kainat.altaf@cs.uni-kiel.de, momin.ali@cs.uni-kiel.de, laura.harms@cs.uni-kiel.de, christian.renner@tuhh.de, olaf.landsiedel@tuhh.de

Abstract

Real-time analysis of underwater acoustic signals is essential for maritime security, environmental monitoring, and wildlife protection. In many real-world deployments, such as remote sensing buoys, autonomous underwater vehicles, and seabed monitoring stations, data must be processed directly on the device due to limited connectivity and energy constraints. These platforms are, for example, tasked with monitoring vessel activity in ecologically sensitive zones, where timely and localized analysis is essential to detect unauthorized presence, enforce protection boundaries, and mitigate acoustic disturbances. Existing approaches mainly focus on coarse ship-type classification or perform individual ship identification at a very limited scale, often restricted to just a handful of vessels. Further, they typically rely on large cloud-based DNNs making them unsuitable for deployment on constrained devices.

To close this gap, we present ShipNN, a resource-efficient DNN model for real-time ship identification from underwater acoustic signals, deployable on resource-constrained microcontrollers. ShipNN achieves 94.1% accuracy in individual ship identification, outperforming, for example, ResNet, a much larger baseline model with merely 92.2% accuracy. ShipNN has 32× fewer parameters, 17× less RAM, and 66× less flash storage than this baseline, demonstrating efficient, real-time operation on off-the-shelf microcontrollers.

CCS Concepts

• **Computing methodologies → Neural networks**; **Supervised learning by classification**.

Keywords

Underwater Acoustics, Ship Identification, Vessel Identification, TinyML, CQT, MFCC, IoT devices

ACM Reference Format:
Kainat Altaf, Momin Ali, Laura Harms, Christian Renner, and Olaf Landsiedel. 2026. ShipNN: On-Device, Ship Identification from Underwater Noise. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748522.3779787

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779787

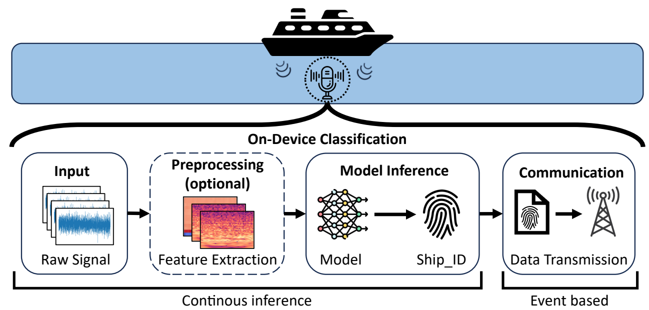


Figure 1: Overview of ShipNN. ShipNN captures the raw audio signal from the ship and optionally pre-processes it to generate spectrograms. A microcontroller runs a lightweight neural network that analyzes the audio or extracted features directly on the device. The neural network identifies the ship and triggers communication, communicating ship identifications to, for example, onshore facilities.

1 Introduction

The ocean is vital for life on Earth for regulating climate, producing oxygen, providing food, sustaining marine biodiversity, and supporting human well-being [14]. Protecting marine ecosystems from threats such as unauthorized fishing and noise pollution in environmentally sensitive areas is essential for preserving ocean health. Similarly, effective maritime traffic management is important for ensuring safe shipping, combating illegal fishing activities, and safeguarding critical infrastructure such as harbors, underwater pipelines, and data cables.

Conventional maritime monitoring technologies, including the Automatic Identification System (AIS), remote sensing, radar, and optical sensors, each have significant limitations. AIS can provide a ship's position, course, and speed, but is vulnerable to data inaccuracies, manipulation, and system failures [11]. Cloud cover and darkness hinder remote sensing via satellite imagery, while poor visibility and camouflage conditions reduce the effectiveness of optical sensors such as cameras [29, 20]. Radar performs well in adverse weather but struggles to differentiate ship types and has limited range [17]. In addition to these methods, modern monitoring systems deploy underwater hydro-acoustic sensors that capture ship noise across distances of several kilometers [2]. These systems transmit the data to onshore facilities for analysis. Continuous audio streaming consumes substantial bandwidth, and many rural or underwater deployments lack the connectivity required to support uninterrupted transmission.

Neural network-based methods have shown strong performance in classifying ship noise into broad vessel categories, such as tugboats or tankers [18, 37, 21], and for identifying a small number of individual ships [9], often using vision models like ResNet [16] or VGGNet [34] to process spectrograms of underwater sounds. However, these deep learning models require substantial computational resources and memory, making them unsuitable for deployment on resource-constrained embedded devices. Moreover, classification into general vessel types is only a first step; the more demanding task of identifying individual ships remains largely unexplored, particularly in the context of embedded, on-device inference.

To address this gap, we present ShipNN, a lightweight deep neural network model and system architecture designed for real-time identification of individual ships from live underwater acoustic signals, see Figure 1. Built on a reduced version of MobileNetV3-Small [19], ShipNN operates within the strict memory and storage limits of microcontroller devices. Despite its compact size, ShipNN achieves 94.1% accuracy. It operates with only 156 kB of RAM and 634 kB of flash memory, and processes one second of input in 604 ms. To enable energy-efficient deployment, ShipNN employs an event-triggered communication strategy that transmits only predictions to onshore facilities. Each prediction is just 8 bytes, ensuring timely ship detection while keeping wireless communication lightweight and energy-conserving.

This paper makes the following contributions:

- We present ShipNN, a lightweight NN model[1] for real-time identification of individual ships based on their acoustic signals. To our knowledge, this is the first work to achieve large-scale ship identification on embedded platforms.
- Our quantized model is 32× smaller than ResNet, a baseline classification model, in terms of both parameters and memory footprint. It requires only 156 kB of RAM and 634 kB of flash memory, making it suitable for microcontroller-based platforms.
- ShipNN transmits predictions instead of raw data and contacts onshore facilities only when a ship is detected, conserving energy and bandwidth.
- With an inference latency of just 604 ms per second of input, ShipNN enables real-time on-device identification of ships.

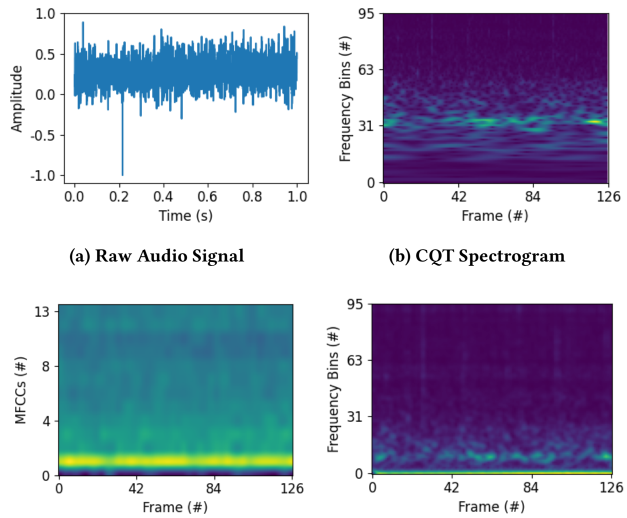
The remainder of this paper is structured as follows: Section 2 presents the required background of ship identification and feature extraction, Section 3 discusses related work, and Section 4 introduces the design of ShipNN. Then, Section 5 evaluates ShipNN on a state-of-the-art dataset, and Section 6 concludes the paper.

2 Background

This section introduces acoustic ship identification using acoustic signals and describes common feature extraction techniques for this task.

2.1 Acoustic Ship Identification

Acoustic ship identification relies on identifying distinctive sound patterns produced by vessel machinery (e.g., propellers and engines), commonly recorded by underwater hydrophones. Unlike transponder-dependent systems like AIS, these sound patterns serve as passive, practically tamper-resistant fingerprints [38]. However, underwater sound propagation varies significantly due to factors such as depth, salinity, temperature, and ambient noise, making it challenging to distinguish ship-specific signals reliably.



(a) Raw Audio Signal (b) CQT Spectrogram

(c) MFCC Spectrogram (d) Gamma Spectrogram

Figure 2: Acoustic signal of a passenger ship recorded at a distance of 2–4 km from the hydrophone (sampling rate: 32 kHz), shown alongside its corresponding feature representations: (a) raw audio signal, (b) Constant-Q Transform (CQT) spectrogram, (c) Mel-Frequency Cepstral Coefficients (MFCC), and (d) Gammatone filterbank response.

2.2 Feature Extraction

For classifying audio signals, they are either processed as raw signals or as frequency-domain representations, such as CQT [8]. Each offers distinct advantages for ship identification, as we discuss in §2.2.1 and §2.2.2.

2.2.1 Raw Audio. Using raw waveform data as input allows deep learning architectures to learn feature representations directly, without relying on predefined transformations. While unprocessed audio provides the model with fine-grained waveform details, learning directly from such high-dimensional data requires large models.

2.2.2 Frequency Domain Representations. Transforming audio signals into frequency-domain representations reveals how energy distributes across time and frequency, allowing models to focus on perceptually and structurally important features while reducing input dimensionality. Among them, Mel-Frequency Cepstral Coefficients (MFCCs) describe the overall shape of the sound by grouping energy into the audible frequency range, reflecting how the human ear hears. Unlike other frequency domain representations, MFCCs don't preserve raw frequency bins; they transform them into a compact set of decorrelated coefficients, the Mel-Frequency Cepstral

[1] ShipNN Github Repository: https://github.com/ds-kiel/ShipNN

Coefficients that summarize audible energy patterns. This abstraction prioritizes mid-range frequencies and compresses less audible extremes. Constant Q-Transform (CQT) spreads frequency bins in a way that highlights mid to low-frequency patterns that ships often produce, making it especially useful for identifying them. Gammatone filterbank focuses on fine details in both frequency and timing, picking up subtle sound details, but tends to include extra noise.

Collectively, these structured representations reduce temporal complexity and enable neural networks to learn patterns more efficiently with fewer parameters [33], see Figure 2. This makes them particularly suitable for embedded scenarios, where both efficiency and accuracy are critical.

2.3 Microcontroller Memory Constraints

Unlike mobile or desktop platforms, a microcontroller (MCU) has strongly limited memory. It stores model weights in flash, while activations and buffers must fit into RAM, which becomes the main bottleneck during inference [30]. Compact architectures and quantization reduce overhead and enable deployment under tight hardware constraints [6]. These optimizations are essential for accurate and efficient ship identification on embedded devices, where MCUs are preferred due to power and cost limits [26].

2.4 Neural Network Architectures for MCUs

Ship classification, which typically involves categorizing vessel types, most often uses ResNet as the backbone architecture [18, 37, 21]. However, its high computational demands make it non-deployable on resource-constrained devices. For such devices, the most commonly used classification models are MCUNet [26] and MobileNet [19], which are optimized for efficiency under resource constraints. For example, MobileNetV3-Small combines depthwise separable convolutions with squeeze and excitation modules, balancing the trade-off between accuracy and model size to suit memory-limited devices, but can only be deployed on MCUs when scaled down. MCUNet offers a lightweight alternative built around inverted residual blocks, enabling efficient inference on MCUs with limited memory and compute capacity. In this paper, we use scaled-down MobileNetV3-Small as the underlying architecture for ShipNN.

3 Related Work

In this section, we discuss today's approaches to acoustic signal identification of underwater noise. These include traditional methods, machine learning, and deep learning models.

3.1 Traditional Processing Methods

Much of the initial research in the field focuses on hand-crafted features and traditional signal-processing methods for feature extraction and classification [8, 5]. Common approaches such as the Fourier Transform (FT), Wavelet Transform (WT), and CQT extract time-frequency domain features from ship-generated noise and other underwater sounds. Further, principal component analysis and canonical correlation analysis enable dimensionality reduction and feature fusion [24]. Compared to today's machine learning models, their dependence on handcrafted features and vulnerability to noise limit their effectiveness in real-world scenarios [1].

3.2 Classification using ML and Deep Learning

Classical machine learning methods for underwater acoustic classification, such as SVMs, HMMs, and GMMs, struggle with nonlinear patterns [2]. In contrast, modern deep learning handles such complex signals well: Hybrid CNN-LSTM models show strong results in sonar classification tasks [37], and transfer learning [27] with pre-trained networks like VGG16 improves results in data-scarce settings. Attention-based transformers enhance classification by capturing both local and global acoustic patterns [13], with spectral transformers handling long-range dependencies [25]. While achieving strong performance, these models (a) focus on classification into general vessel types and (b) are too resource-intensive for real-time inference on constrained devices.

3.3 Identification using ML and Deep Learning

Only a few studies focus on vessel identification. Kamal et al. [23] use deep belief networks for target identification with hierarchical feature learning. SAE-softmax [3] applies sparse autoencoders to extract invariant features from spectral data, refining them with stacked autoencoders and softmax classifiers. An attention-based neural network processes pressure spectrograms to distinguish underwater targets [37]. Similarly, a ResNet-based Siamese network compares pairs of acoustic signals for ship identification [12]. Despite their contributions, these approaches typically focus on a narrow set of types, often limited to two or three vessel types, and depend on large models that cannot run on edge devices. Such designs are impractical for deployment in edge environments where latency, memory, and power constraints are critical.

4 Design

ShipNN introduces a lightweight DNN for real-time ship identification in remote marine environments deployable on low-power embedded hardware. Designed for energy-efficient operation, ShipNN performs ship identification in four main steps, see Figure 1. (1) An underwater hydrophone captures live ship acoustics. (2) The system either feeds these raw signals directly to the DNN model or passes them through an optional feature extraction module that converts them into compact representations such as spectrograms. (3) A quantized, resource-efficient DNN model deployed on a microcontroller then identifies individual ships. (4) Upon detecting a ship, the system communicates the sighting to, for example, an on-shore station.

In the following, we first discuss feature extraction in §4.1 and the neural network architecture of ShipNN in §4.2. Next, we introduce data augmentation to increase model robustness for real-world acoustic data in §4.3. In §4.4, we present system details, including quantization and deployment of the model on a microcontroller. Finally, in §4.5 we discuss our communication model.

4.1 Feature Extraction

After the hydrophone captures acoustic signals, ShipNN processes acoustic signals by either feeding raw waveforms directly into the model or converting them into spectrograms through an optional feature extraction stage. Raw audio contains fine-grained temporal features that lightweight models struggle to learn effectively. Capturing these details requires large models that are not deployable on the MCU. At a sampling rate of 32 kHz, the signal produces 32,000 data points per second. We use spectrograms, e.g., CQT, gammatone, and MFCC (see §2.2), to transform this time-domain signal into a compact frequency-domain format, reducing the input size to under 12,000 data points per second, a 62.5% decrease. This transformation enables faster inference on resource-constrained devices.

We generate the spectrograms using a 125-sample hop size and Hann window [15], balancing temporal resolution with smooth spectral transitions. This configuration yields 95 frequency bins spanning 18 to 4186 Hz for both CQT and gammatone, effectively capturing ship-relevant acoustic patterns while suppressing high-frequency noise. For MFCCs, we extract 13 coefficients, a standard [36] in acoustic analysis, chosen to capture perceptually relevant frequency components while discarding higher-order terms often encoding noise or redundancy. We pad MFCC inputs with the mean value to match the dimensions of other feature types, ensuring compatibility across models.

Among the different spectrogram types, CQT tends to highlight mid to low-frequency harmonic patterns typical of ship sounds. In contrast, MFCC offers a perceptual summary that supports moderate performance, while gammatone features capture subtle auditory details but introduce redundancy and noise sensitivity. Figure 2 plots an example energy distribution of a ship, illustrating the properties of the different types of spectograms. CQT (Figure 2b) reveals clear harmonic structure in mid to low frequencies, MFCC (Figure 2c) shows broad spectral contours, and gammatone (Figure 2d) appears more diffuse and less structured. From this, we identify CQT as our best candidate for feature extraction in ship noise. Our evaluation in §5 underlines this observation.

4.2 Neural Network Models

Given the limitations of embedded hardware, such as constrained memory, processing power, and energy resources, deploying large-scale architectures like ResNet is infeasible. Therefore, we need identification models that are not only lightweight to run on edge devices but also ensure good accuracy. To establish suitable alternatives, we experiment with both MCUNet and the full MobileNetV3-Small architecture. While these networks achieve competitive accuracy, their memory footprint and inference latency exceed the practical limits of microcontroller-class devices.

Motivated by these insights, we explore variants of MobileNetV3-Small. As we show in our evaluation in Section 5.2, the MobileNetV3-Small variant with 25% width offers a more favorable trade-off. It combines compact size and real-time inference latency while maintaining strong classification performance. To our knowledge, earlier work does not explore such lightweight architectures for ship identification. We therefore consider ResNet, MCUNet, and full MobileNetV3-Small as baselines and adopt MobileNetV3-Small (25%) as the core model of ShipNN due to its balanced efficiency and accuracy, without introducing any architectural modifications.

4.3 Data Augmentation

Additionally, to enhance robustness under limited data conditions, we employ spectrogram augmentation [28], which randomly masks frequency bins and time frames in the input spectrogram. Practically, this augmentation doubles the data per ship. As we show in our evaluation in Section 5.3, this helps the model generalize better to unseen acoustic conditions where vessel signals may be partially masked or distorted in real-world environments.

4.4 Deployment

To deploy ShipNN on low-power edge devices, we apply post-training full-integer (INT8) linear quantization, which reduces model size and computational complexity. This step converts 32-bit floating-point models into 8-bit fixed-point representations, lowering memory usage and speeding up inference. We run the quantized model with LiteRT for Microcontrollers [10] and accelerate integer operations on Cortex-M7 DSPs through CMSIS-NN. As a result, ShipNN enables real-time detection directly in the water with live data on buoys and underwater sensors.

4.5 Communication

Once the model generates predictions, ShipNN transmits ship identification results to onshore facilities using LoRa [4]. LoRa provides low-energy, low-cost connectivity over seawater distances exceeding 22km [22], making it a practical alternative to satellite links. Its spreading factor controls a trade-off between range, latency, and power use; we set it to 10 to ensure real-time transmission over longer distances with minimal energy overhead. To further conserve energy, ShipNN triggers communication only when it detects a ship, avoiding unnecessary or repetitive transmissions and reducing network load.

5 Evaluation

In this section, we experimentally evaluate ShipNN. We begin by providing an overview of the training details in §5.1, including the dataset and evaluation metrics. Next, in §5.2, we analyze the identification accuracy of ShipNN against the baselines and compare the performance of different feature extraction methods. In §5.3, we evaluate the effect of data augmentation. Next, we assess on-device performance in §5.4, followed by an evaluation of event-based communication in §5.5. Finally, in §5.6, we discuss our results and provide insights into the DNN behavior.

5.1 Training Details

We train ShipNN to identify individual ships using underwater acoustic recordings from the VTUAD dataset [7]. The following sections detail the dataset structure, implementation setup for model training, deployment, and evaluation on embedded hardware.

5.1.1 Dataset. We use the Vessel Type Underwater Acoustic Data (VTUAD) dataset, collected by Ocean Network Canada [7] in the Strait of Georgia between June 24 and November 3, 2017. An iCListen AF hydrophone, deployed 147m below sea level, captures each 1-second uncompressed WAV recording at a 32kHz sampling rate.

VTUAD labels vessels into five categories: cargo, tanker, tug, passenger, and background. It also includes Maritime Mobile Service Identity (MMSI) tags, which we use for ship-level identification. To ensure sufficient data per class, we select 100 ships from the full set of 223, each with 156 unique samples.

Table 1: Comparison of ShipNN and baselines MCUNet (MCN), MobileNetV3-Small (MBN), and ResNet (RN) across different preprocessing techniques, including individual methods and their combinations. "All Spec." refers to the combination of all spectrogram-based preprocessing methods. The table reports accuracy, precision, and recall for each setting, with the best accuracy values shown in bold. CQT captures important features and yields strong accuracy across most models, while Gamma features result in poor performance. The CQT + MFCC combination shows the best overall results.

Preprocessing	Accuracy				Precision				Recall			
	ShipNN	MCN	MBN	RN	ShipNN	MCN	MBN	RN	ShipNN	MCN	MBN	RN
Raw	88.7	88.2	89.4	89.1	80.9	88.8	65.9	92.6	80.5	88.1	89.8	91.8
CQT	91.1	56.1	91.4	91.9	94.4	73.9	93.4	96.7	94.1	56.1	92.9	96.3
MFCC	64.2	82.9	75.0	85.5	84.1	83.4	76.4	90.0	83.3	82.9	74.9	89.7
Gamma	42.8	49.8	44.3	63.4	78.0	65.1	62.9	89.5	73.0	49.8	44.3	88.6
CQT + Gamma	88.7	54.7	84.6	89.4	93.3	71.6	87.5	95.5	93.0	54.7	84.5	95.3
CQT + MFCC	**92.7**	**92.0**	**91.7**	**92.2**	92.8	92.5	92.8	96.4	92.8	92.0	91.7	96.2
MFCC + Gamma	80.3	81.1	88.9	90.1	90.3	81.8	89.5	93.8	89.3	81.1	88.9	93.6
All Spec.	91.3	90.8	87.4	89.1	92.0	91.5	88.9	96.4	91.2	90.8	87.3	96.1

The dataset defines three scenarios based on inclusion and exclusion radii to isolate individual vessels and vary signal-to-noise conditions. The inclusion radius defines the area containing a single vessel, while the exclusion radius ensures no other vessels are present in the surrounding zone between the inclusion and exclusion boundaries. Scenario S_1 (2km inclusion, 4km exclusion) covers 24% of the data, S_2 (3km inclusion, 5km exclusion) covers 43%, and S_3 (4km inclusion, 6km exclusion) covers the remaining 33%.

5.1.2 Implementation Details. We implement all models in TensorFlow except for MCUNet, which is trained using PyTorch. All models use the Adam optimizer, a batch size of 32, and an 80/10/10 train-validation-test split. ShipNN converges at 500 epochs, while MCUNet and MobileNetV3-Small converge at 200 epochs, all with a learning rate of 5e−4. ResNet converges at 100 epochs with a learning rate of 1e−4.

To demonstrate real-world feasibility, we deploy the models on the STM32H747XIH6 [35], utilizing only the Cortex-M7 with 2MB Flash and 1MB RAM. We measure system energy consumption with the Nordic Power Profiler Kit II [32], and we use the SX1276 LoRa module [4] to estimate communication latency and energy cost when transmitting predictions to onshore facilities.

5.2 Feature Extraction and Accuracy

We begin our evaluation by comparing the performance of our feature extraction methods (CQT, MFCC, Gamma), their combinations (e.g., CQT and Gamma combined), and the raw data on ShipNN and our baselines, namely MCUNet, MobileNetV3-Small, and ResNet, see Table 1. CQT generally provides the best results for ShipNN, achieving 91.1% accuracy, while MFCCs yield moderate performance (64.2%) and gammatone features significantly degrade accuracy (42.8%). This trend replicates across the baselines as well and aligns with the characteristics of the methods, see Section 4.1.

Feature combinations further emphasize the contrast among features. Configurations including CQT consistently outperform others. CQT and MFCCs get the best result for all models: ShipNN with 92.7% accuracy, MCUNet 92.0%, MobileNetV3-Small with 91.7%, and ResNet with 92.2%. In contrast, adding gammatone features

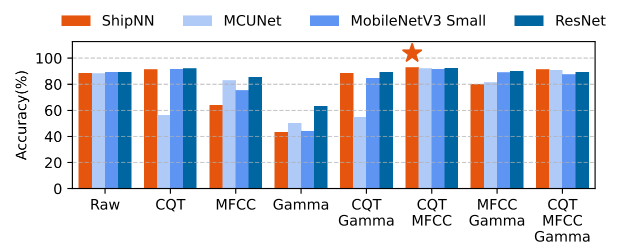


Figure 3: Comparison of classification accuracy across different preprocessing methods. ShipNN (351k parameters) achieves performance comparable to, and in several cases surpassing MCUNet, MobileNetV3-Small, and ResNet baselines. Despite its significantly smaller model size, ShipNN consistently maintains competitive accuracy, showing particular strength on combined CQT+MFCC features.

tends to reduce accuracy across all models. Notably, when combining all three feature types, ShipNN still maintains high performance (91.3%), surpassing larger baselines such as ResNet and MobileNetV3-Small with 89.1% and 87.4% identification accuracies, respectively. Using raw waveforms as input provides a stronger baseline than gammatone in the majority of cases (88.7% for ShipNN, 88.2% for MCUNet, 89.4% for MobileNetV3-Small, and 89.1% for ResNet), but still underperforms settings with combined spectrogram features. These findings highlight that lightweight models, when paired with effective feature extraction, can rival and even surpass larger architectures, underscoring the importance of choosing robust preprocessing methods (see Figure 3).

5.3 Dataset Augmentation

To improve robustness under real-world conditions such as background noise and signal distortion, we further apply dataset augmentation to ShipNN as a complementary step to strengthen generalization. Augmentation improves the performance of the best-performing feature configurations, albeit with moderate gains, see Table 2. For instance, accuracy increases from 91.1% to 92.6% for

Table 2: Dataset augmentation improves ShipNN performance across spectral inputs, especially with CQT+MFCC, enhancing generalization and robustness under limited data. "All Spec." refers to the combination of all spectrogram-based preprocessing methods.

Preprocessing	Aug	Accuracy	Precision	Recall
CQT	×	91.1	94.4	94.1
	✓	**92.6**	92.9	92.6
CQT+MFCC	×	92.7	92.8	92.8
	✓	**94.1**	94.4	94.1
All Spec.	×	91.3	92.0	91.2
	✓	**93.9**	94.3	93.9

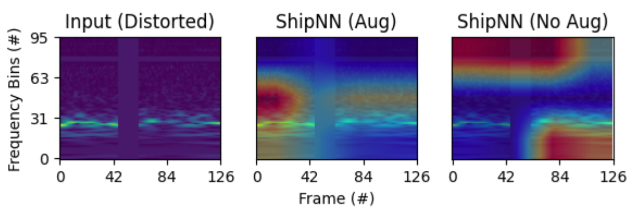


Figure 4: Visualization of ShipNN predictions with and without data augmentation for a distorted input signal. With augmentation, ShipNN attends to discriminative spectrogram regions and correctly detects ships; while without augmentation, it focuses on irrelevant areas and often misclassifies.

CQT, from 92.7% to 94.1% for CQT+MFCC, and from 91.3% to 93.9% when we combine all features.

Visual interpretation using Grad-CAM [31] (see Figure 4) reveals how ShipNN processes acoustic features with and without data augmentation. The ShipNN's model trained with data augmentation accurately identifies ships even from distorted or noisy signals, attending to the most discriminative spectrogram regions. In contrast, the model trained without augmentation often misclassifies such samples, focusing instead on irrelevant or noisy areas. These visualizations indicate that data augmentation leads to more stable and meaningful feature representations, improving overall generalization under real-world acoustic variability.

5.4 On-Device Performance

After evaluating feature extraction and data augmentation, we next evaluate the on-device performance of our models. For this, we choose the best configuration of each model in terms of accuracy and model size, using the combined CQT+MFCC feature extraction for all experiments. ResNet requires 42 MB of flash memory, which makes it far too large for embedded deployment. Even when quantized, its flash requirement remains 10.5 MB, still exceeding microcontroller limits. MobileNetV3-Small is significantly smaller than ResNet; however, it still exceeds the available memory of typical Cortex-M7 microcontrollers like the STM32H747, making it non-deployable. In contrast, MCUNet (709 kB flash, 508 kB RAM)

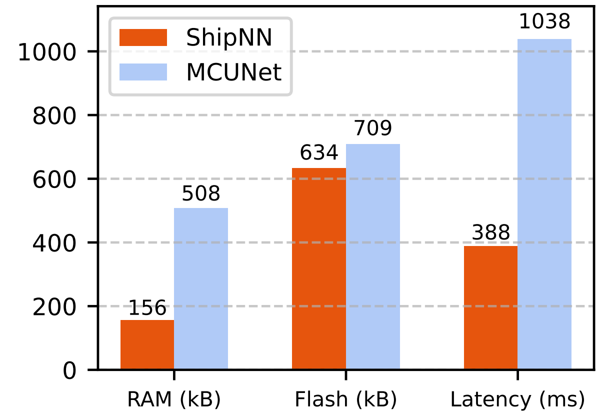


Figure 5: Comparison of memory footprint and inference latency for ShipNN and MCUNet on STM32H747XIH6 microcontroller. ShipNN demonstrates a balanced profile across all metrics, combining moderate memory usage with acceptable latency suitable for real-time inference. MCUNet uses slightly more memory than ShipNN, and its inference latency exceeds one second (for a one second recording), making it impractical for real-time classification tasks.

Table 3: Computational complexity of ShipNN in terms of parameters and size of the model: ShipNN and MCUNet are significantly smaller than ResNet and MobileNetV3-Small, therefore deployable on microcontrollers.

Model	Params (M)	Flash (kB)	Deployable
ResNet	11.3	42000	×
ResNet Quantized	11.3	10500	×
MobileNetV3-Small	1.63	2950	×
MCUNet	0.575	709	✓
ShipNN	0.351	634	✓

and ShipNN (634 kB flash, 156 kB RAM) are compact enough for microcontroller deployment, as shown in Table 3.

Beyond memory footprint, latency and energy efficiency further differentiate the two deployable models. Both MCUNet and ShipNN use CQT+MFCC preprocessing, which takes 216 ms to compute. Following preprocessing, MCUNet completes inference in 1038 ms, resulting in a total end-to-end latency of 1254 ms per second of input. Thus, it does not fulfill real-time requirements. In contrast, ShipNN processes 1 second of input in 388 ms, yielding a total end-to-end latency of 604 ms for both feature extraction and DNN processing, comfortably within real-time constraints. Similarly, energy consumption measurements highlight that ShipNN is significantly more efficient (see Figure 6). Although ShipNN exhibits a slightly higher peak current (14.03 mA vs. 12.01 mA for MCUNet), its faster execution reduces total energy consumption by more than 50% (32.1mJ vs. 66.6mJ).

Considering memory usage, latency, and energy consumption collectively, ShipNN provides the best results, making it the most

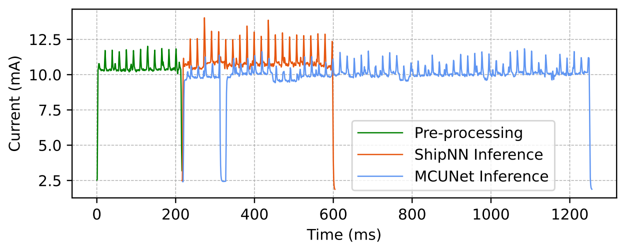


Figure 6: Comparison of current consumption during pre-processing and inference for MCUNet and ShipNN on STM32H747XIH6 microcontroller. ShipNN achieves the same task in less than half the time (604 ms vs. 1254 ms) while consuming over 50% less total energy (32.1mJ vs. 66.6mJ). Despite a slightly higher peak current (14.03mA vs. 12.01mA), ShipNN 's faster execution and lower overall energy demand make it far more efficient for edge deployment than MCUNet.

suitable solution for real-time, on-device ship identification on resource-constrained microcontrollers.

5.5 Communication Performance

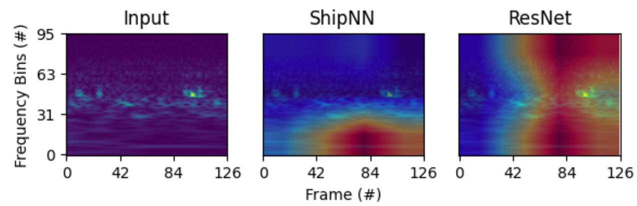
To enable real-time ship detection, ShipNN must transmit identified vessels to onshore facilities. We use LoRa for wireless communication and measure transmission time to evaluate communication latency. Each prediction carries only 8 bytes of data (4 bytes for the ship's ID and 4 bytes for a timestamp), making transmission efficient over constrained channels. Our measurements show latency rising from 61.7 ms at spreading factor 7 (SF7) to 1482.7 ms at SF12, with energy consumption increasing from 5.90 mJ to 141.89 mJ. This reflects the fundamental trade-off in LoRa: a higher spreading factor extends transmission range but increases latency and energy cost. We select SF10, with 370.6 ms latency and 35.4 mJ energy, as the optimal configuration, balancing sufficient range with real-time responsiveness and energy efficiency for sustainable long-term monitoring.

5.6 Discussion and Insights

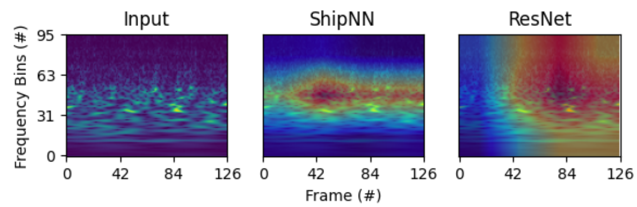
We now shift from results to interpretation, analyzing how our models behave and perform under real-world constraints. We examine how ShipNN's neural network processes acoustic features, how performance varies with ship–hydrophone distance, and how data augmentation makes ShipNN robust for real-world scenarios.

GradCAM visualizations offer insight into how ShipNN and ResNet attend to different acoustic cues. Figure 7 presents examples from a cargo ship at a distance of 2 to 4 km and a tugboat at 3 to 5 km. ShipNN focuses on distinct frequency bands and time windows, isolating propulsion harmonics while ignoring background noise. For the cargo ship, attention concentrates on lower frequencies where engine signatures dominate. For the tugboat, it shifts to mid-frequency bands to capture noisier propulsion patterns. In contrast, ResNet activates broadly across the spectrum, indicating less precise localization.

This focused attention translates into consistent performance across ship–hydrophone distances. The 2 to 4 km range contains the fewest audio samples, yet ShipNN achieves the highest accuracy at 91.6% (see Figure 7), outperforming all baselines. Such robustness under data scarcity and acoustic variability underscores its suitability for embedded maritime monitoring, where consistent inference is critical despite limited resources.



(a) Cargo ship signal at 2–4 km distance



(b) Tugboat signal at 3–5 km distance

Figure 7: CQT spectrogram of the live input signal (left), with corresponding attention relevance maps for ShipNN (middle) and ResNet (right). The heatmaps highlight the regions most influential for the model's predictions, showing differences in how each model localizes important spectral features.

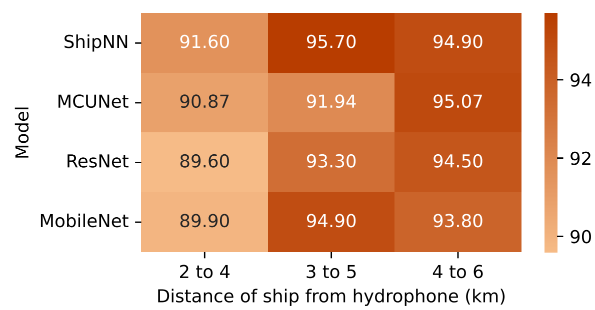


Figure 8: Accuracy of detection models across increasing ship to hydrophone distances. Despite having significantly fewer parameters (351k), ShipNN consistently outperforms larger models such as MCUNet (575k), MobileNetV3-Small (1.63M), and ResNet (11.3M). Its accuracy remains stable across all distance ranges, demonstrating that even with a lightweight architecture, ShipNN delivers competitive performance for long-range ship detection on resource-constrained devices.

6 Conclusion

In this work, we introduce ShipNN, a resource-efficient deep neural network and system architecture for on-device ship identification.

When quantized, its small size enables deployment on off-the-shelf microcontrollers such as ARM Cortex M7. Our evaluation on the VTUAD dataset demonstrates ShipNN's effectiveness, achieving 94.1% accuracy, outperforming ResNet's 92.2% while using 32× fewer parameters. ShipNN requires only 156 kB of RAM and 634 kB of flash while achieving a real-time inference latency of 604 ms for one second of input data. ShipNN is robust to handle real-world signal distortions, supports deployment in offshore platforms, and enables practical applications such as maritime traffic monitoring, fleet management, and environmental enforcement under severe energy and connectivity constraints.

Acknowledgments

This project has received funding from the Federal Ministry for Economic Affairs and Climate Action under the Marispace-X project grant no. 68GX21002E and the Federal State of Schleswig-Holstein under the INSYST grant no. 22023008.

References

[1] Silvia Adrián-Martínez, Manuel Bou-Cabo, Ivan Felis, Carlos D Llorens, Juan Antonio Martínez-Mora, María Saldaña, and Miguel Ardid. 2015. Acoustic signal detection through the cross-correlation method in experiments with different signal to noise ratio and reverberation conditions. In *Ad-hoc Networks and Wireless: ADHOC-NOW 2014 International Workshops, ETSD, MARSS, MWaoN, SecAN, SSPA, and WiSARN, Benidorm, Spain, June 22–27, 2014, Revised Selected Papers 13*. Springer, 66–79.

[2] Muhammad Azeem Aslam, Lefang Zhang, Xin Liu, Muhammad Irfan, Yimei Xu, Na Li, Ping Zhang, Zheng Jiangbin, and Li Yaan. 2024. Underwater sound classification using learning based methods: a review. *Expert Systems with Applications*, 255, (Dec. 2024), 124498.

[3] Xu Cao, Xiaomin Zhang, Yang Yu, and Letian Niu. 2016. Deep learning-based recognition of underwater target. In *2016 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 89–93.

[4] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. 2016. Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios. In *IEEE Wireless Communications* number 5. Vol. 23. IEEE, 60–67.

[5] Donald G. Childers, David P. Skinner, and Robert C. Kemerait. 1977. The cepstrum: a guide to processing. *Proceedings of the IEEE*, 65, 1428–1443, 10.

[6] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. 2019. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 3009–3018.

[7] Lucas Domingos, Phillip Skelton, and Paulo Santos. 2022. Vtuad: vessel type underwater acoustic data. (2022). https://dx.doi.org/10.21227/msg0-ag12.

[8] Lucas CF Domingos, Paulo E Santos, Phillip SM Skelton, Russell SA Brinkworth, and Karl Sammut. 2022. A survey of underwater acoustic data classification methods using deep learning for shoreline surveillance. *Sensors*, 22, 6, 2181.

[9] Xiaoyang Du and Feng Hong. 2024. Qiandaoear22: a high-quality noise dataset for identifying specific ship from multiple underwater acoustic targets using ship-radiated noise. *EURASIP Journal on Advances in Signal Processing*, 2024, 1, 96.

[10] Google AI Edge. 2025. Litert: high-performance runtime for on-device ai. https://ai.google.dev/edge/litert. Accessed: 2025-10-02. (2025).

[11] Ties Emmens, Chintan Amrit, Asad Abdi, and Mayukh Ghosh. 2021. The promises and perils of automatic identification system data. *Expert Systems with Applications*, 178, 114975.

[12] Haotian Fang, Xiaowei Zhang, and Ji Xu. 2023. Underwater acoustic target identification via siamese network. *OCEANS 2023 - Limerick, OCEANS Limerick 2023*. ISBN: 9798350332261.

[13] Sheng Feng and Xiaoqian Zhu. 2022. A transformer-based deep learning network for underwater acoustic target recognition. *IEEE Geoscience and Remote Sensing Letters*, 19.

[14] 2025. Goal 14: oceans - united nations sustainable development. Accessed: 2025-10-05. United Nations, (2025). https://www.un.org/sustainabledevelopment/oceans/.

[15] Fredric J. Harris. 1978. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66, 1, 51–83.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

[17] Xiaoyang He, Xiaolong Chen, Xiaolin Du, Xinghai Wang, Shuwen Xu, and Jian Guan. 2025. Maritime target radar detection and tracking via dtnet transfer learning using multi-frame images. *Remote Sensing*, 17, 5, 836.

[18] Feng Hong, Chengwei Liu, Lijuan Guo, Feng Chen, and Haihong Feng. 2021. Underwater acoustic target recognition with a residual network and the optimized feature extraction method. *Applied Sciences 2021, Vol. 11, Page 1442*, 11, (Feb. 2021), 1442, 4, (Feb. 2021). https://www.mdpi.com/2076-3417/11/4/1442/htm%20https://www.mdpi.com/2076-3417/11/4/1442.

[19] Andrew Howard et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1314–1324.

[20] Zishuo Huang, Qinyou Hu, Qiang Mei, Chun Yang, and Zheng Wu. 2021. Identity recognition on waterways: a novel ship information tracking method based on multimodal data. *The Journal of Navigation*, 74, 6, 1336–1352.

[21] Muhammad Irfan, Zheng Jiangbin, Shahid Ali, Muhammad Iqbal, Zafar Masood, and Umar Hamid. 2021. Deepship: an underwater acoustic benchmark dataset and a separable convolution based autoencoder for classification. *Expert Systems with Applications*, 183, (Nov. 2021), 115270.

[22] Nikola Jovalekic, Dragan Marić, and Srdjan Krčo. 2018. An experimental evaluation of lora communication range over seawater. In *2018 26th Telecommunications Forum (TELFOR)*. IEEE, 420–423.

[23] Suraj Kamal, Shameer K Mohammed, PR Saseendran Pillai, and MH Supriya. 2013. Deep learning architectures for underwater target recognition. In *2013 Ocean Electronics (SYMPOL)*. IEEE, 48–54.

[24] Bernice Kubicek, Ananya Sen Gupta, and Ivars Kirsteins. 2022. Canonical correlation analysis as a feature extraction method to classify active sonar targets with shallow neural networks. *The Journal of the Acoustical Society of America*, 152, (Nov. 2022), 2893–2904, 5, (Nov. 2022). https://pubmed.ncbi.nlm.nih.gov/36456259/.

[25] Peng Li, Ji Wu, Yongxian Wang, Qiang Lan, and Wenbin Xiao. 2022. Stm: spectrogram transformer model for underwater acoustic target recognition. *Journal of Marine Science and Engineering 2022, Vol. 10, Page 1428*, 10, (Oct. 2022), 1428, 10, (Oct. 2022). https://www.mdpi.com/2077-1312/10/10/1428/htm%20https://www.mdpi.com/2077-1312/10/10/1428.

[26] Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. 2020. Mcunet: tiny deep learning on iot devices. *Advances in neural information processing systems*, 33, 11711–11722.

[27] Amirmohammad Mohammadi, Tejashri Kelhe, Davelle Carreiro, Alexandra Van Dine, and Joshua Peeples. 2025. Cross-domain knowledge transfer for underwater acoustic classification using pre-trained models. In *OCEANS 2025 Brest*. IEEE, 1–6.

[28] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: a simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

[29] Marco Reggiannini et al. 2024. Remote sensing for maritime traffic understanding. *Remote Sensing*, 16, 3, 557.

[30] Swapnil Sayan Saha, Sandeep Singh Sandha, and Mani Srivastava. 2022. Machine learning for microcontroller-class hardware: a review. *IEEE Sensors Journal*, 22, 22, 21362–21390.

[31] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 618–626.

[32] Nordic Semiconductor. 2020. Power profiler kit ii. Accessed: 2025-10-06. (2020). https://www.farnell.com/datasheets/3449681.pdf.

[33] Roneel V Sharan, Hao Xiong, and Shlomo Berkovsky. 2021. Benchmarking audio signal representation techniques for classification with convolutional neural networks. *Sensors*, 21, 10, 3434.

[34] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

[35] STMicroelectronics. 2019. Stm32h747xi/g - dual 32-bit arm® cortex®-m7 up to 480mhz and -m4 mcus. Accessed: 2025-10-06. (2019). https://www.st.com/resource/en/datasheet/stm32h747ag.pdf.

[36] Wenbo Wang, Sichun Li, Jianshe Yang, Zhao Liu, and Weicun Zhou. 2016. Feature extraction of underwater target in auditory sensation area based on mfcc. *2016 IEEE/OES China Ocean Acoustics Symposium, COA 2016*, (Aug. 2016). ISBN: 9781467399784.

[37] Xu Xiao, Wenbo Wang, Qunyan Ren, Peter Gerstoft, and Li Ma. 2021. Underwater acoustic target recognition using attention-based deep neural network. *JASA Express Letters*, 1, (Oct. 2021), 106001, 10, (Oct. 2021). /asa/jel/article/1/10/106001/218605/Underwater-acoustic-target-recognition-using.

[38] Eduardo Zurek, Margarita R Gamarra, José R Escorcia, Carlos Gutierrez, Henry Bayona, Roxana P Pérez, and Xavier García. 2014. Acoustic fingerprint recognition using artificial neural networks. *Polibits*, 50, 63–73.