Classification of Vessel Types by Means of Machine Learning

Konstantinos Karatziotis[1], Karoline Gussow[1], Viktoriia Boichenko[1], Christian Kanarski[1], Bastian Kaulen[1], Frederik Kühne[1], Marco Driesen[1], Finn Röhrdanz[1], Lukas Schirmer[1], Ralf Burgardt[1], Gerhard Schmidt[1]

[1] *Digital Signal Processing and System Theory, Department of Electrical and Information Engineering, Kiel University,*
Email: {koka, kars, vib, chk, bk, frk, madr, finr, lusc, rabu, gus}@tf.uni-kiel.de

Abstract

Monitoring and identification of vessels and underwater objects are essential tasks for securing maritime infrastructures such as ports and shipping routes. Moving vessels produce characteristic sound waves based on vessel-specific parameters such as propeller type and speed, which can be detected by passive sound navigation and ranging (SONAR) systems. Currently, experienced human SONAR operators classify these sounds, often with impressive success rates, but automated decision-making approaches can assist new operators or even allow autonomous systems to perform classification independently. In our study, we developed a machine-learning-based vessel-classification system that uses underwater acoustic recordings to identify vessels and determine their class. In this process, the recordings are pre-processed with time-frequency analysis and demodulation methods to extract features. These features are used as input to train and evaluate a convolutional neural network (CNN) specialized for classification. To maximize performance, the CNN undergoes several training cycles with different configurations and will be evaluated compared to an extended version.

Acoustic Signatures of Vessels

Typically, a vessel has a propeller with several blades for movement. When these blades rotate in the water, regions of high and low pressure are created on their surface, leading to the formation of bubbles. These bubbles are unstable and collapse, generating acoustic noise known as cavitation noise. The acoustics of a vessel $s(t)$ can be described as follows [1]:

$$s(t) = \left[1 + \sum_{k=0}^{K-1} m_k \sin(2\pi k f_0 t + \phi_k)\right] v_c(t) + v_a(t). \quad (1)$$

This equation describes the cavitation noise $v_c(t)$, which is modulated by the fundamental frequency f_0. The variable m_k represents the modulation index of the k-th harmonic, while $v_a(t)$ denotes ambient noise. The first harmonic of the fundamental frequency corresponds to the rotational frequency of the propeller and can be extracted through demodulation. Other factors contributing to the vessel's acoustics $s(t)$ include machinery noise, vessel size, speed and maneuvering actions.

Data

Most published studies on the classification of vessel types based on their emitted acoustics are often self-recorded and not publicly available. Currently, there are two large publicly available databases: ShipsEar [2] and DeepShip [3]. For training the neural network, only the ShipsEar database was used. The ShipsEar database is provided by the University of Vigo and was recorded using a passive SONAR system between 2012 and 2013 at the harbor of Vigo, located in the northwest of Spain on the Atlantic Ocean. The recordings range in length from 15 seconds to 10 minutes and include 11 different vessel types, categorized by size. For data acquisition, up to three hydrophones were used, which were attached vertically to a buoy. In shallow waters, one or two hydrophones were used for recording. The data was digitized using a 24-bit analog-to-digital converter with a sampling rate f_s of 52.734 kHz. The database classifies vessels into the following categories:

- Class A: Trawlers
- Class B: Motorboats, sailboats
- Class C: Ferries
- Class D: Cruise ships, ro-ro vessels
- Class E: No vessel, background noise

Preprocessing

For classifying the data using machine learning methods, the data is preprocessed to emphasize relevant features. The neural network receives these features as input and determines which vessel type could have generated them. Some recordings in the database have a duration of several minutes. Such long signals are not necessary for training, which is why the data is divided into multiple segments. This also increases the amount of data available for training, validation, and testing. For preprocessing, records with a length of 2.6 s are used. Additionally, a butterworth high-pass filter with a cutoff frequency of $\omega_{\text{cut}} = 10$ Hz is applied to the data to remove DC components. Two main methods are used to extract features, namely time-frequency analysis and modulation analysis, which provides information about the propeller frequency. Time-frequency analysis is commonly used for analyzing acoustic signals and is performed using the Short-time Fourier transform (STFT). The STFT

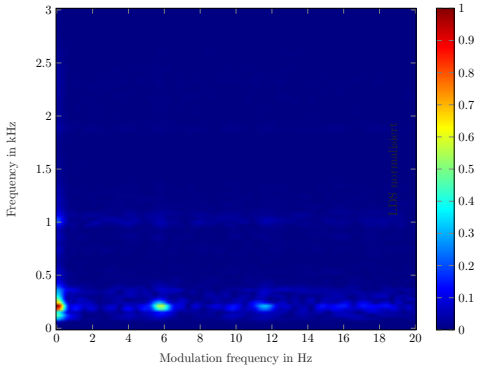

Figure 1: Modulation analysis spectrogram of a tugboat with $N_{\text{mod}} = 512$.

is given by

$$V(\mu, k) = \sum_{n=0}^{N_{\text{STFT}}-1} v(kR+n)w(n)e^{-j\mu\frac{2\pi}{N_{\text{STFT}}}n}, \quad (2)$$

which divides a signal into multiple segments using a window function $w(n)$. Subsequently, the Discrete Fourier transform (DFT) of length N_{STFT} is computed for each windowed segment. The frequency and time resolution are influenced by the DFT length N_{STFT} and the size of the window function $w(n)$. The result of an STFT is typically visualized as a spectrogram, which displays frequency over time. The spectrogram of a vessel primarily contains frequency components in the lower range, from 100 Hz to 1 kHz, appearing as constant lines. A second method for extracting vessel acoustics features is modulation analysis, which can be used to analyze the rotational frequency of the propeller and its number of blades. At the first stage, the input signal $v(n)$ is modulated by several complex exponential terms, corresponding to a frequency shift in the frequency domain,

$$v(n)e^{-j\mu_i\frac{2\pi}{N_{\text{mod}}}n} \quad (3)$$

where i denotes the i-th subband. The modulation leads to a complex-valued signal:

$$\overline{v}_i(n) = \overline{v}_{i,\text{Re}}(n) + j\overline{v}_{i,\text{Im}}(n). \quad (4)$$

After modulation, the signal of the i-th subband is filtered by the low-pass filter $h_{i,u}$:

$$\overline{v}_{i,\text{LP}}(n) = \sum_{u=0}^{U-1} \overline{v}_i(n)h_{i,u}. \quad (5)$$

Due to the modulation in the previous step, filtering the signal with a low-pass filter can be interpreted as band-pass filtering. After band-pass filtering, the envelope of each subband $\tilde{v}_i(n)$ is computed by taking the squared magnitude:

$$\tilde{v}_i(n) = |\overline{v}_{i,\text{LP}}(n)|^2. \quad (6)$$

The frequencies of a vessel are primarily located in the lower frequency range. The envelopes of the subbands are therefore smoothed with a low-pass filter g_u and then downsampled:

$$\tilde{v}_{i,\text{LP}}(n) = \sum_{u=0}^{U-1} \tilde{v}_i(n-u)g_u, \quad (7)$$

$$\hat{v}_i(n) = \tilde{v}_{i,\text{LP}}(Rn). \quad (8)$$

The spectrum of the envelope, which contains information about the propeller frequency of a vessel, is obtained by computing the DFT of length N_{mod} of each subband, windowed by $w(n)$:

$$O_{i,N_{\text{mod}}}(\mu) = \text{DFT}_{N_{\text{mod}}}\{\hat{v}_i(n)w(n)\}. \quad (9)$$

Similar to the STFT, modulation analysis can be visualized in a spectrogram, where one axis represents the subbands and the other axis represents the modulation frequency. Such a spectrogram is referred to as a modulation analysis spectrogram and is shown in Figure 1.

Data Variation

The performance of the neural network is heavily influenced by the features of the data and their representation. To emphasize the lower frequencies, the magnitudes of the spectrograms of the STFT and the modulation analysis are modified accordingly. One approach is to transform the frequency axis into the mel scale using

$$m = 2595\,\text{Mel}\log_{10}\left\{1 + \frac{f}{700\,\text{Hz}}\right\}, \quad (10)$$

which is inspired by human auditory perception. Another method for emphasizing lower frequency bands is to transform the frequency axis into a logarithmic scale. This is achieved by computing a logarithmic sequence based on a chosen base and step size. For each value in this sequence, the corresponding amplitude value from the linear frequency axis is assigned. If the value lies between two points, it is approximated using linear interpolation. For logarithmic scaling, bases of 2 and 10 are used. Spectrograms with different frequency scales are shown in Figure 2. All spectrograms are normalized to a range between 0 and 1 using

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}. \quad (11)$$

Additionally, the spectrograms differ in how their magnitude is computed. The magnitude is computed using absolute values (abs), squared magnitudes (abs squared), and decibels (dB).

Network Layout

Due to the two-dimensional nature of the data resulting from the preprocessing techniques, a CNN is used as the foundation for extracting features from the spectrograms. The architecture of the CNN is given in Table 1. It begin s with a convolutional layer consisting of 32 kernels, each with a size of 5×5, and a rectified linear unit (ReLU) activation function. This is followed by a max-pooling layer with a size of 2×2 for dimensionality reduction and a dropout layer with a dropout rate of 0.5

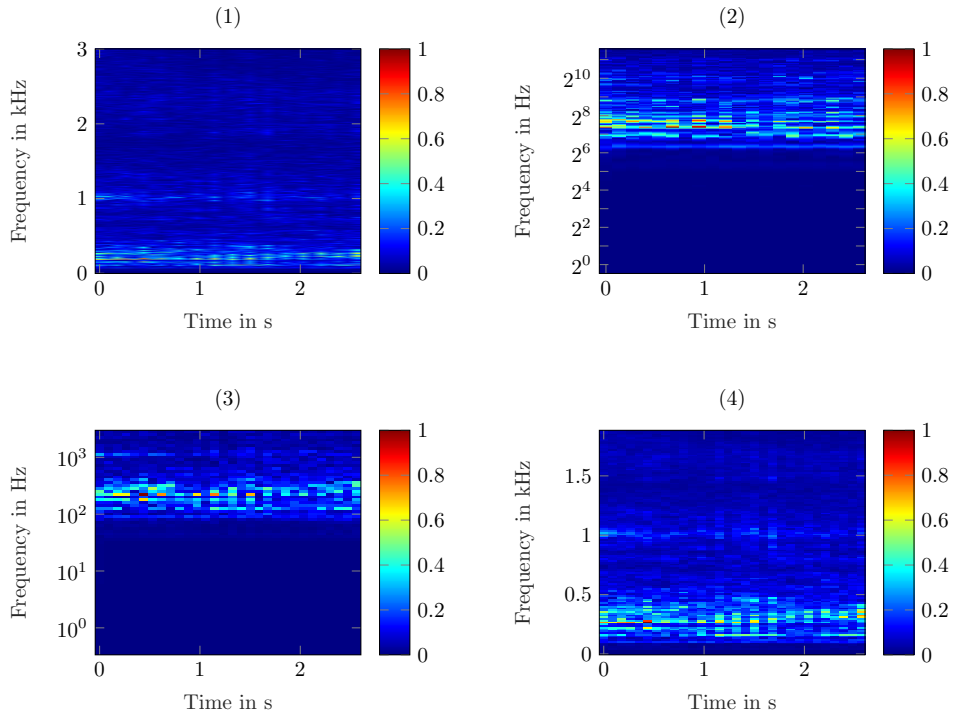

Figure 2: STFT with linear (1), mel (4), and logarithmic frequency scaling of bases 2 (2) and 10 (3).

to prevent overfitting. This sequence of convolutional, max-pooling, and dropout layers is repeated, with the second convolutional layer containing 64 kernels. Next, a flattening layer is applied to convert the data into a one-dimensional vector, followed by another dropout layer. The final output of the CNN is a dense layer with five neurons and a softmax activation function, providing the probability distribution across all five labels.

Table 1: CNN architecture used for classification.

Layer	Kernels	Size	Neurons	Act.
Conv.	32	5×5	/	ReLU
MaxP.	/	2×2	/	/
Drop.	/	/	/	/
Conv.	64	5×5	/	ReLU
MaxP.	/	2×2	/	/
Drop.	/	/	/	/
Flatt.	/	/	/	/
Dense	/	/	128	ReLU
Drop.	/	/	/	/
Dense	/	/	5	SoftMax

In addition to the standard CNN, the neural network architecture is extended with additional features. The general architecture of the neural network is depicted in figure 3, where the red-marked part represents the extension adding the additional features.

Four additional features are incorporated into the network, extracted from the modulation analysis. These features include the propeller frequency, identified as the maximum value in a certain part of the modulation analysis spectrogram, the frequency band of the propeller

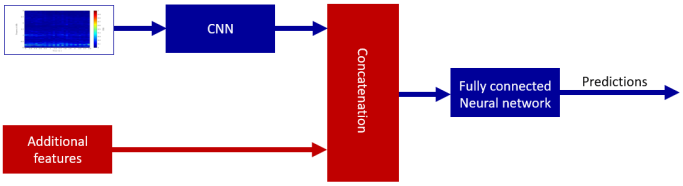

Figure 3: Architecture of the neural network with its extension.

frequency, its magnitude, and the input-to-noise ratio (INR). To calculate the INR of the signal $v(n)$, the noise $b(n)$ is estimated in the spectrum:

$$\widehat{S}_{vv}(\mu, n) = |V(\mu, k)|^2, \tag{12}$$

$$\overline{S_{vv}(\mu, k)} = \beta\, \overline{S_{vv}(\mu, k-1)} + (1-\beta)\, \widehat{S}_{vv}(\mu, k), \tag{13}$$

$$\widehat{S}_{bb}(\mu, k) = \begin{cases} \max\left\{S_{\min},\, \widehat{S}_{bb}(\mu, k-1)\right\} \Delta_{\text{inc}}, \\ \qquad \text{if } \overline{S_{vv}(\mu, k)} > \widehat{S}_{bb}(\mu, k-1), \\ \max\left\{S_{\min},\, \widehat{S}_{bb}(\mu, k-1)\right\} \Delta_{\text{dec}}, \\ \qquad \text{else.} \end{cases} \tag{14}$$

Subsequently, the spectrum is calculated by element-wise division of the input spectrum by the estimated noise spectrum if a defined threshold δ is exceeded:

$$\text{INR}(\mu, k) = \begin{cases} \frac{\widehat{S}_{vv}(\mu,k)}{\widehat{S}_{bb}(\mu,k)}, & \text{if } \widehat{S}_{vv}(\mu, k) > \delta \\ 1, & \text{else.} \end{cases} \tag{15}$$

A scalar value for the INR is obtained by averaging along the frequency μ and time axes n, respectively.

$$a = \frac{1}{K N_{\text{ana}}} \sum_{k=0}^{K-1} \sum_{\mu=0}^{N_{\text{ana}}-1} \text{INR}(\mu, k). \tag{16}$$

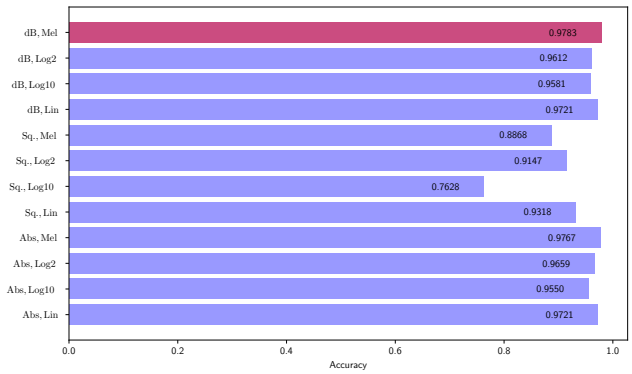

Figure 4: Accuracy of the base model.

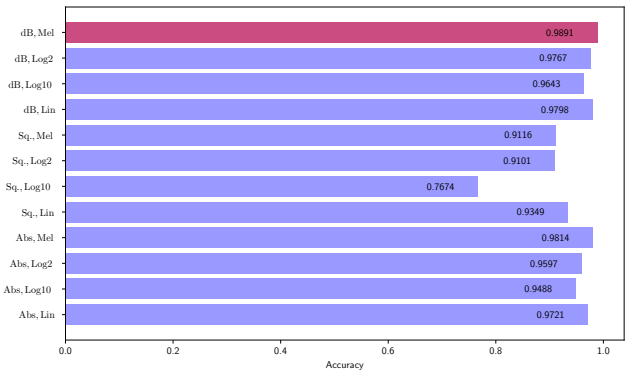

Figure 5: Accuracy of the extended model.

Evaluation

To evaluate the performance of the neural network, the data is split into 60 % training, 20 % validation, and 20 % testing. Care is taken to ensure that all datasets maintain an identical class distribution, allowing for a reasonable comparison of results. The neural network is trained iteratively for 40 epochs with a batch size of 32. Training is conducted using Python and TensorFlow for two network architectures: one consisting solely of the CNN and an extended version with additional features. Throughout training, validation accuracy and loss are monitored using the validation dataset. The validation loss is computed based on entropy:

$$J = -\frac{1}{L}\sum_{l=0}^{L-1} y_l \log(\hat{y}_l), \tag{17}$$

where y_l represents the true label, $\hat{y}_l$ the predicted label, and L the total number of labels. The CNN is trained using various STFT configurations. To simplify notation, the format Power,Scale is introduced. The "Power" index specifies the method of power computation: Abs, Sq, and dB, corresponding to absolute magnitude, squared magnitude, and decibel scaling, respectively. The "Scale" index denotes the frequency scale used: Lin (linear), Log2 (logarithmic base 2), Log10 (logarithmic base 10), and Mel (mel scale). After training and validation, the model's performance is evaluated using the 20 % test dataset. The results for both the base and extended networks are shown in Figure 4 and 5. The extended network shows slightly improved performance, achieving an accuracy of approximately 98.91 % with the dB,Mel configuration, compared to 97.83 % for the base network under the same configuration. Thus, utilizing the architecture of the extended neural network, the error can be reduced by 47.88 %.

Summary and Outlook

Using a CNN with the ShipsEar database, it was demonstrated that underwater vessel recordings can be classified based on vessel size. Preprocessing the data by transforming the time-domain signals into a time-frequency representation yielded promising results, particularly when applying a mel scale to the frequency axis and representing the magnitude in dB. With this configuration, an accuracy of 97.83 % was achieved. The classification performance was further improved by modifying the neural network. The extended model incorporates additional features, which are concatenated after the CNN's flattening layer. Three of these features are extracted from the modulation analysis, while one represents the input-to-noise ratio (INR). This extended architecture increased the accuracy to 98.91 %. A key limitation of the current model is its dependence on the ShipsEar database. Applying the trained network to a different dataset may lead to challenges, as variations in measurement setups can significantly impact the perfomnce of the neural network. To develop a more robust model, it is essential to generate as much diverse training data as possible. However, the availability of suitable data remains a major constraint.

References

[1] Z. Liu, L. Lü, C. Yang, Y. Jiang, L. Huang and J. Du, "DEMON Spectrum Extraction Method Using Empirical Mode Decomposition," 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, Japan, 2018, pp. 1-5, doi: 10.1109/OCEANSKOBE.2018.8559175.

[2] David Santos-Domínguez, Soledad Torres-Guijarro, Antonio Cardenal-López, Antonio Pena-Gimenez, ShipsEar: An underwater vessel noise database, Applied Acoustics, Volume 113, 2016, Pages 64-69, ISSN 0003-682X, https://doi.org/10.1016/j.apacoust.2016.06.008.

[3] Muhammad Irfan, Zheng Jiangbin, Shahid Ali, Muhammad Iqbal, Zafar Masood, Umar Hamid, DeepShip: An underwater acoustic benchmark dataset and a separable convolution based autoencoder for classification, Expert Systems with Applications, Volume 183, 2021, 115270, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.115270.