Regular Article

Underwater Acoustic Target Classification Using Convolutional Neural Network Combined with Continuous Wavelet Transform

Ngoc Anh Tu Nguyen[1], **Van Sang Doan**[2]

[1] Le Quy Don Technical University, Ha Noi, Vietnam
[2] Faculty of Communication and Radar, Vietnam Naval Academy, Nha Trang City, Khanh Hoa, Vietnam

Correspondence: Van Sang Doan, doansang.g1@gmail.com
Communication: received 31 January 2024, revised 15 May 2024, accepted 25 May 2024
Online publication: 01 June 2024, Digital Object Identifier: 10.21553/rev-jec.363

Abstract– **Underwater acoustic target (UAT) classification is a critical task in submarine warfare, enabling sonar operators and commanders to understand the surrounding situation in the operational area. In order to improve the accuracy of UAT classification, a convolutional neural network (CNN) combined with continuous wavelet transform (CWT), so-called CWT-CNN, is proposed in this article. Specifically, signal preprocessing methods such as short-time Fourier transform (STFT), continuous wavelets transform (CWT) and mel-frequency cepstral coefficients (MFCC), which are employed for providing feature maps to the CNN classifier, are taken into account for analysis and comparison. The experiment indicates that the CWT-CNN model achieved the best accuracy of 99.64% compared to other considered methods.**

Keywords– **Convolutional neural network, continuous wavelet transform, short time Fourier transform, mel-frequency cepstral coefficients, underwater target classification.**

1 Introduction

During diving operations, submarines have to monitor the activities of surrounding objects to avoid collisions and detect targets at a far range. The task of target recognition is manually performed by sonar operators [1, 2], who are trained to listen to received sounds. In this way, the accuracy of target classification relies heavily on the operator's experience. Along with the advancement of science and technology, maritime vehicles are rapidly increasing in both quantity and variety, leading a major challenge in target identification. Some machine learning (ML) methods [3] have been proposed and shown effectiveness but still have several limitations. For example, support vector machine (SVM) can accurately classify a dataset by finding a hyperplane (or many hyperplanes) in the space of the data points such that the distance between the two closest points in opposite classes to the hyperplane are maximized. In many cases, data is not easily separable, thus necessitating the establishment of a mathematical penalty for data points lying on the wrong side of the hyperplane. In another work, Liu *et al.* [4] studied a model to enhance the classification of targets using ship radiated signals, where the feature extraction of line spectrum and the SVM classifier were employed. The SVM classification algorithm attained a high accuracy when applying optimized SVM parameters (94%). However, the work only aims at the basic classification situation of binary SVM (with two classes). Therefore, SVM models can face difficulties with large training sets and multi-label or multi-domain classification problems.

In another approach, k-nearest neighbors (K-NN) algorithm allows for a relatively simple classification model that uses a known dataset to classify new data by polling the k closest data points in the known dataset. The new data point is subsequently assigned a classification based on the class that has the highest representation among its k nearest neighbors. The underlying rationale for this approach is straightforward: points belonging to the same class are expected to share similar inherent characteristics. Consequently, the features of data points within the same class should exhibit sufficient similarity, positioning them in proximity to each other. The concept of "nearness" in this context relates to the spatial arrangement of points in the feature space and the distance between them, often quantified by Euclidean distance. As the value of k increases, the classification becomes more resistant to the influence of outlier data points. Therefore, k serves as a hyper-parameter that can be effectively chosen through a validation process. The intuition is that a larger k promotes a more robust classification, less prone to the impact of individual outliers, and this hyper-parameter's optimal value can be determined through validation procedures. K-NN classifiers requires accessing all training samples and computing their differences from classified samples, which burdens computational resources and system memory. For instance, Tong *et al.* [5] proposed a classification experiment on the radiated noise of three types of measured underwater targets. MFCC feature vectors of these targets are extracted, and the K-NN algorithm is employed for classification and identification. Based on the experimental results, it is evident that as the order of the MFCC feature vector increases, the corresponding target recognition accuracy also increases.

However, this also leads to an increase in data volume

and computational complexity. In addition, decision tree (DT) is also a simple algorithm that is widely used and effective for classification tasks. However, the DT model heavily depends on the training data. Even a small change in the dataset, such as adding new data points or modifying existing ones, can significantly alter the structure of the decision tree model [6]. This sensitivity can lead to overfitting, where the model performs well on the training data but poorly on unknown data.

In recent years, a new approach called Deep Learning (DL) has gained attention and research focus. This approach performs sound recognition and classification using spectrogram features, promising high accuracy and gaining widespread usage. For instance, Wei *et al.* [7] investigated the spectrogram characteristics of MFCC in audio signals, combining them with a Deep Neural Network (DNN) to improve the quality of data input for underwater acoustic signal analysis and enhance target identification. Extraction of MFCC characteristic parameters is a method to classify and identify signals inspired by the human auditory perception mechanism. This represents a major research direction in the signal processing of underwater acoustic sensors. Although this approach holds promise for achieving high classification accuracy, it still faces several limitations as highly sensitive in noisy audio. The drawbacks of MFCC features in noisy environments rely on many factors include spectrum estimation methods, design of effective filter banks, and the number of chosen features, which are also affecting the complexity of the audio recognition systems.

Jin *et al.* [8] introduced a novel framework that applied the LOFAR spectrum for pre-processing (using STFT) to preserve key features and selected a neural network modified LENET to improve classification performance. However, STFT typically use a fixed window size for signal processing, which may not be optimal for capturing all frequency components in underwater acoustic signals. Doan *et al.* [9] developed a CNN model named UATC-DenseNet for classifying underwater acoustic targets from raw audio signals. This work asserted that the improvement in performance was attributed to model optimized the utilization of features represented across multiple layers through appropriate use of skip-connections. Nevertheless, when the input data is raw in time domain, it can lead to a decreased ability to distinguish between useful signals and noise. This can lead to the model being unable to effectively extract important features, resulting in reduce accuracy in classification and overall performance. Based on those promises, the paper proposes a model that combines signal transformation method using the Continuous Wavelet Transform (CWT) with Convolutional Neural Network (CNN) to increase the accuracy of underwater target classification, overcoming the limitations of other traditional models and providing valuable decision support for sonar operators. The following sections will present the proposed method, evaluate the model and conclusions.

2 Research Method

2.1 System Model and Preprocessing

2.1.1 System Model: Currently, most submarines are equipped with passive sonar systems [10] that are responsible for listening to sound emitted from various underwater sources. Sonar operators rely on their hearing, knowledge, and experience to interpret and make decisions about the type of sound source. Building upon this foundation, this study investigates a device that connects to the submarine's sonar system through an audio port, as illustrated in Figure 1. The device has task to capture the audio signals using a sound card, perform preprocessing, and apply a neural network for sound classification. Since the device only captures signals from the audio port, it does not interfere with the operation of the existing sonar system. The classification results will assist sonar operators in making decisions. It serves as independent corroborative information to enhance reliability and confidence for the operators.

2.1.2 Preprocessing: In the audio signal processing, the Fourier Transform (FT) is an essential tool as it serves as a bridge between the time domain and the frequency domain representation of a signal [11, 12]. Representing a signal in the frequency domain provides insights into the distribution of energy across different frequencies, which can sometimes offer advantages over the time domain. The Fourier Transform of a signal $x(t)$ is defined as follows

$$X(\omega) = \int_{-\infty}^{+\infty} e^{-i\omega t} x(t) dt, \tag{1}$$

where ω is the angular frequency. However, the Fourier Transform is only effective when the frequency spectrum is stationary, meaning that the signal frequencies do not vary over time. In reality, most signals are inherently non-stationary. The spectrum indicates which frequencies are present in the signal but not always appear. To address this issue, the Short-Time Fourier Transform (STFT) was developed [13]. It overcomes the issue by dividing the signal into small portions with equal-sized segments (possibly overlapping), such that each segment can be considered as a stationary signal. The Fourier Transform is then applied to each segment. The general formula for the STFT is as follows

$$X(\omega, t_0) = \int_{-\infty}^{+\infty} \mathrm{w}(t) e^{-i\omega t} x(t) dt, \tag{2}$$

where t_0 is the time, which is redefined based on the time corresponding to each frame of the signal, and $\mathrm{w}(t)$ is the window function. Therefore, the STFT reveals the frequencies at specific time instances in the time domain. However, the STFT is constrained by the uncertainty principle concerning high and low frequency components in the signal. It means that the choice of the width of segments must be appropriate, as a smaller width provides better time resolution but poorer frequency resolution, and vice versa. Wavelet

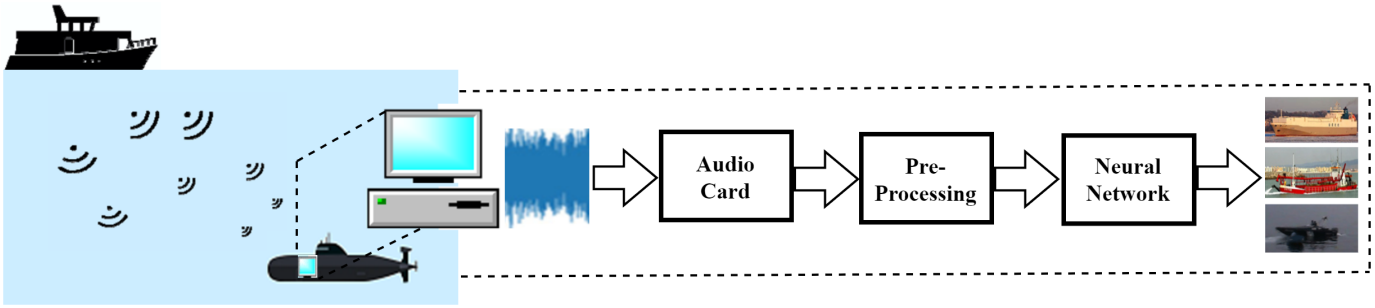


Figure 1. Passive sonar system for underwater acoustic signal classification.

transform is the next solution to overcome the limitations of STFT. The key difference in wavelet transform is the ability to analyze signals at multiple scales or with different resolutions [14]. The approach involves constructing a basic wavelet function $\psi(t)$ based on transformations and filtering operations. It is characterized by several parameters such as scale and position. The construction of the Wavelet function involves the following steps:

- Select a "mother wavelet" function, that serves as the basis for the wavelet transform. The choice of the "mother wavelet" depends on the specific application and desired properties. Commonly used mother wavelets include the haar wavelet, daubechies family wavelets, and morlet wavelet.
- The "mother wavelet" function is then transformed and adjusted by scaling and translation operations to create a family of "daughter wavelets". Scaling involves stretching or compressing the function in the time domain, while translation involves shifting the function along the time axis.

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{|s|}}\psi(\frac{t-\tau}{s}); s \neq 0, \tau \in R, \tag{3}$$

in which, s represents the dilation or compression of the signal (frequency), while the parameter is the translation or shift of the signal in the time domain. The continuous wavelet transform is defined as follows

$$X(s,\tau) = \int_{-\infty}^{+\infty} \psi_{s,\tau}(t)x(t)dt. \tag{4}$$

When performing wavelet transform on a signal, we obtain a multi-scale representation of the signal, enabling us to examine the signal at different levels of details. This helps capture the signal's features at different scales and resolutions. In this case, the transformed coefficients s and τ are arranged in a scaleogram form. This scaleogram can then be used as an input image for classifiers such as CNN, SVM, K-NN, and DT.

2.2 Proposed CNN Model

In this paper, the proposed CNN model's structure is depicted in Figure 2. It can be observed that the preprocessed signal serves as the input to the CNN model. The input data for STFT and CWT features, originally specified as a 64-by-64 matrix in indexed image format, are converted to RGB (red-green-blue) image format, resulting in sizes of 64×64×3. The same process is applied to MFCC features, resulting in a size of 64×32×3. Meanwhile, the raw signal has a size of 4096×1. Following the input layer is a normalization layer, which aims to normalize the input values to enhance the rate of convergence and stability of the deep learning network. The normalization process is carried out by subtracting the mean value, μ, and dividing by the standard deviation $\sqrt{\sigma^2+\varepsilon}$ using the formula as follows

$$\text{Norm}(\text{x}) = \frac{\text{x}-\mu}{\sqrt{\sigma^2+\varepsilon}}. \tag{5}$$

The two quantities, μ and σ^2, are estimated based on the statistical data of the input data cluster (mini-batch) for the corresponding training iteration. In this case, a very small constant $\varepsilon > 0$ is added to the formula to prevent division by zero, ensure numerical stability during the calculation process, even when the estimated variance σ^2 equals zero.

After the normalization step, the signal will be fed into a convolutional block consisting of three main layers: the convolutional layer (conv), the max-pooling layer (maxpool), and the activation layer (ReLU: Rectified Linear Unit). In Figure 4, the input is a batch of m signal samples with dimensions (H_0, W_0, C_0), where C_0 represents the number of channels in a two-dimensional (2D) matrix. In this case, applying n filters of size (f, f, C_0) will yield an output array of four dimensions (m, H_1, W_1, n). The values of matrix (H_1, W_1) are computed through the convolutional operation.

The max pooling layer then reduces the spatial dimensions (height and width) of the feature maps while preserving the most prominent features. It partitions each feature map into non-overlapping regions and retains the maximum value within each region. This down-sampling operation helps to reduce the computational complexity and improve the translation invariance of the network. Following the max pooling layer, the activation layer applies the ReLU activation function element-wise to the feature maps. The ReLU function sets negative input values to zero and keeps positive values unchanged, introducing non-linearity into the network. This non-linear activation aids in modeling complex relationships and enables the network to learn more expressive representations. One thing to note when using the ReLU function is that initializing the filter values in the convolutional layer

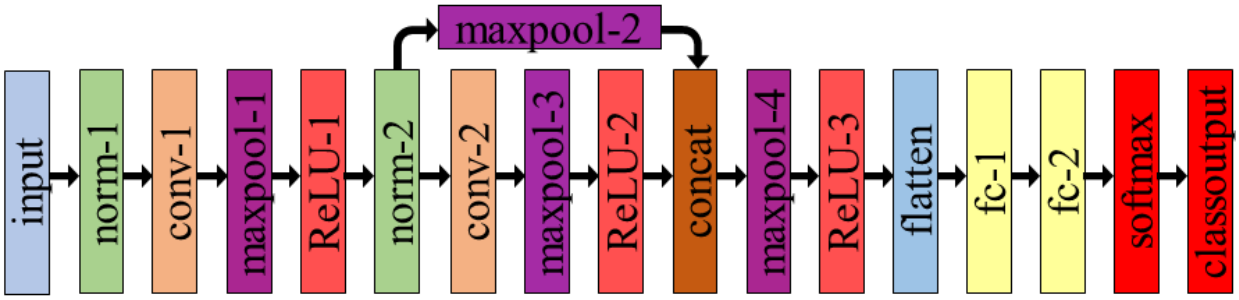


Figure 2. Structure of the proposed CNN model.

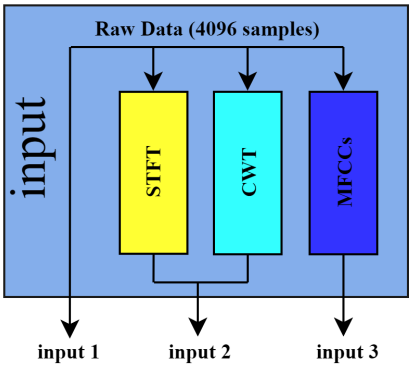


Figure 3. Input block

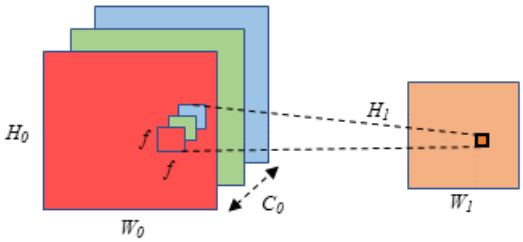


Figure 4. Operation of multi-channel convolution.

or having a high learning rate can lead to the issue of "dead neurons", where the values of network nodes become zero, rendering the computation in subsequent layers meaningless. The concatenated layer refers to the process of combining the outputs of multiple layers along a particular axis (dimension). This operation is often used to merge information from different sources or parts of the network. Based on the above-mentioned mechanism, the CNN model structure in Figure 2 is sequentially implemented through the layers: norm-1, conv-1, maxpool-1, ReLU-1, norm-2, maxpool-2, conv-2, maxpool-3, ReLU-2, concat, maxpool-4, ReLU-3. The number of hidden layers in the network helps improve accuracy, but it also increases the complexity of computations and training time. Therefore, it is necessary to choose an appropriate balance between efficiency and workload. Following the convolutional blocks, the feature maps are flattened into a vector (flatten) and passed through fully connected layers (fc-1, fc-2). These layers connect every neurons from the previous layer to the subsequent layer, similar to a standard neural network. Fully connected layers extract high-level representations from the learned features and enable

Table I
Structural Parameters for the Proposed CNN Model

Layer	Filter	Output 1	Output 2	Output 3
input	-	4096 × 1 × 1	64 × 64 × 3	64 × 32 × 3
norm-1	-	4096 × 1 × 1	64 × 64 × 3	64 × 32 × 3
conv-1	32*(7×7)	4096 × 1 × 32	64 × 64 × 32	64 × 32 × 32
maxpool-1	3×3	2048 × 1 × 32	32 × 32 × 32	32 × 16 × 32
ReLU-1	-	2048 × 1 × 32	32 × 32 × 32	32 × 16 × 32
norm-2	-	2048 × 1 × 32	32 × 32 × 32	32 × 16 × 32
conv-2	32*(7×7)	2048 × 1 × 32	32 × 32 × 32	32 × 16 × 32
maxpool-3	3×3	1024 × 1 × 32	16 × 16 × 32	16 × 8 × 32
ReLU-2	-	1024 × 1 × 32	16 × 16 × 32	16 × 8 × 32
maxpool-2	3×3	1024 × 1 × 32	16 × 16 × 32	16 × 8 × 32
concat	-	1024 × 2 × 32	16 × 32 × 32	16 × 16 × 32
maxpool-4	3×3	512 × 1 × 32	8 × 16 × 32	8 × 8 × 32
ReLU-3	-	512 × 1 × 32	8 × 16 × 32	8 × 8 × 32
flatten	-	16384 × 1	4096 × 1	2048 × 1
fc-1	-	128 × 1	128 × 1	128 × 1
fc-2	-	10 × 1	10 × 1	10 × 1
softmax	-	10 × 1	10 × 1	10 × 1
classoutput	-	10 × 1	10 × 1	10 × 1

complex pattern recognition. The fc-2 layer is set up with 10 neurons at the output, corresponding to the 10 target classes for classification. The 10-element vector then goes through the softmax layer, which computes the probability of each target class among the 10 predefined classes. Finally, this probability is used to predict the target class for all inputs. Through the refinement and experimentation process, the CNN model with the parameters in Table I was selected for the underwater sound classification.

3 Results and Analysis

3.1 Dataset Description and Training Option

The training dataset used in this study was obtained from the ShipsEar data source [15]. The researchers utilized the digitalHyd SR-1 recording device to passively capture audio segments emitted by different types of ships along the Atlantic coastline of northwestern Spain. The dataset consists of a total of 47 recordings, ranging from 15 seconds to 10 minutes of time duration, including 9 ship types and 1 environmental noise type, as summarized in Table II. After removing the empty segments without any information, the remaining audio recordings are divided into 5000 smaller files, each with a size of 4096 samples at a sampling rate of 44.1 kHz for each marine target. This segmentation ensures that each file captures a manageable segment of the sound for efficient feature extraction. Feature extraction techniques, such as STFT, CWT, or MFCC, are then applied to these segments to convert the raw

Table II
The Statistics of Audio Recordings for Each Marine Target

Class	Recordings	Class	Recordings
DredGer (DG)	5	PilotShip (PL)	2
FishBoat (FB)	4	RORO (RR)	5
MusselBoat (MB)	5	SailBoat (SB)	4
OceanLiner (OL)	7	TrawLer (TL)	1
TugBoat (TB)	2	Noise (N)	12

Table III
The CNN Model Combined with Preprocessing Methods

Features	RAW SIG.	STFT	CWT	MFCC
No. Par.	2.1M	580.7K	580.7K	318.5K
Acc. (%)	98.18	98.55	99.64	97.58
Train. Time	81m 50s	62m 12s	66m 54s	53m 28s
Pred. Time (ms)	10.6±2.4	12.4±1.8	33.6±5.3	15.2±3.2

audio into a numerical representation suitable for the CNN model. The distribution of these files is as follows:

- 80% of files (4000 files) are allocated for training.
- 20% of files (1000 files) are allocated for validation and testing.

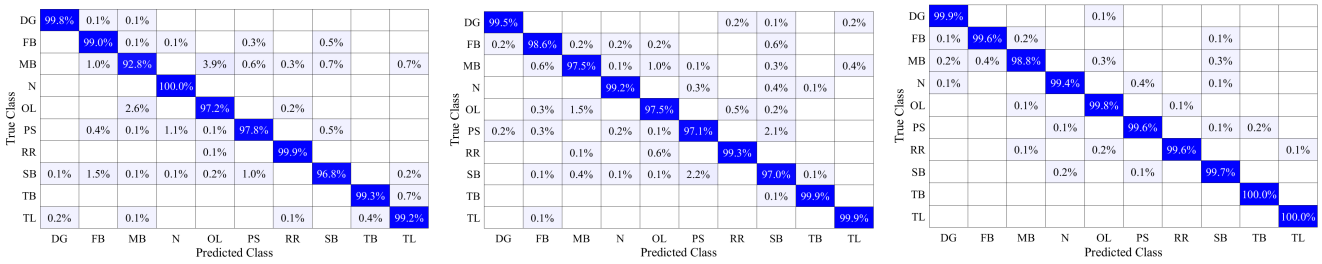
The experimental results were conducted using MATLAB 2022b, running in the GPU execution environment of NVIDIA GeForce RTX 3060Ti. Some parameters need to be configured for the training process. Firstly, the number of epochs, or training cycles, must be chosen appropriately as it significantly influences the model's accuracy. A model needs a sufficiently large number of epochs to learn the general structure of the data. However, when the number of epochs continues to increase, and the accuracy on the validation set plateaus or even decreases, it is an indication that the model has reached optimal performance and further training is unnecessary. In this study, experimentation was conducted with 40 epochs, resulting in an accuracy of 99.64%, which outperformed training with 30 epochs (99.57%) and 50 epochs (99.63%). Secondly, the mini-batch size, representing the number of data points used in each weight update, affects the model's performance and learning speed. A larger size accelerates computation as operations can be performed in parallel, but it demands more memory resources. Conversely, a smaller size has the opposite effect. Experiments revealed that with the same number of epochs, varying the mini-batch size (16, 32, 64) led to a gradual reduction in training time. Simultaneously, accuracy declined, with a mini-batch size of 16 proving to be the optimal choice. Finally, the initial learning rate is set at 0.001, and the Stochastic Gradient Descent with Momentum (SGDM) optimizer is employed. The learning rate is a crucial hyper-parameter that determines the size of the steps taken during the optimization process. A higher learning rate may cause the model to converge quickly but risks overshooting the optimal solution, while a lower learning rate might result in slow convergence or getting stuck in local minima. The choice of 0.001 suggests a moderate learning rate, and SGDM, which combines aspects of both Stochastic Gradient Descent and momentum, is a popular optimizer choice for its ability to navigate complex optimization landscapes effectively. Fine-tuning these hyper-parameters ensures the model converges efficiently and achieves optimal performance during the training process.

3.2 Results and Discussions

In the first experiment, the proposed model was combined with difference preprocessing methods, including waveform (raw signal data), STFT, CWT, and MFCC to classify underwater acoustic signals. As a result, the performance in terms of classification accuracy and training time is shown in Table III.

According to Table III, the feature extraction method using CWT used for input data achieves the highest accuracy (99.64%). It indicates that the CWT method enhances the extracted features of the signal, thereby increasing the classification accuracy. The STFT method achieves the second-best performance with an accuracy of 98.55%. The process of using raw data (waveform) with the CNN method also yields relatively good results (98.18%). The MFCC method achieves the lowest performance with an accuracy of 97.58%. Considering the complexity (number of parameters and computation cost) as one of the most crucial factors in real-time system. In Table III, we also provide additional information on the number of learnable parameters and the average prediction time. The model CNN-RAW SIG. has the highest parameters (2.1M, where M stands for millions), leading to the longest training time (Train. Time) of 81m 50s. However, due to the absence of signal preprocessing, this model has the shortest prediction time (Pred. Time: 10.6ms). CNN-CWT model has the longest (Pred. Time: 33.6ms), which is also a major limitation of the proposed model. Figure 6 depicts the training process curves with input samples having different features.

The confusion matrix, as represented in Figure 5, shows the performance of different preprocessing methods. In the matrix, the rows represent the actual targets, and the columns represent the targets predicted by the model. All values are based on the number of testing instances. The misclassification rates are primarily high for target types such as FishBoat, MusselBoat, OceanLiner, PilotShip, and SailBoat. The notable instances of mispredictions occurring outside the main diagonal are primarily concentrated on the target pairs: FishBoat and MusselBoat, PilotShip and SailBoat, MusselBoat and OceanLiner. The most significant number of mispredictions occurs between MusselBoat and OceanLiner, and vice versa, in Figure 5(a). The corresponding values decrease in Figure 5(b) and in Figure 5(c). However, the number of misclassifications increases gradually within the group of FishBoat and MusselBoat. The confusion trend can be explained by the similar in signal characteristics among the ship types within the same group. Overall, the CWT method achieves the highest average accuracy. Though, for each target type, different preprocessing methods may have more suitable features. Based on theory and experimentation, it can be concluded that CWT is a feasible feature extraction tool for audio data in the task of



(a) CNN-Waveform (b) CNN-STFT (c) CNN-CWT

Figure 5. Confusion matrix for different preprocessing methods.

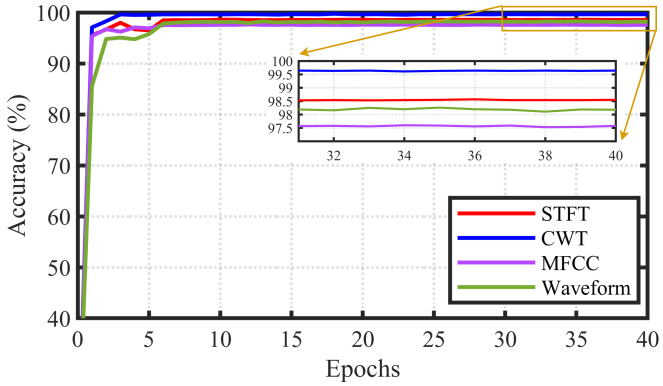


Figure 6. Learning curves for different features.

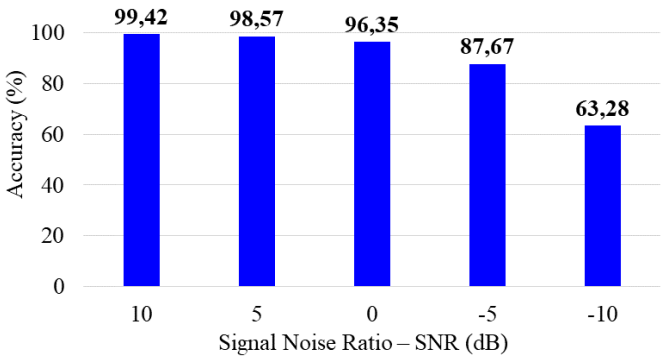


Figure 7. Classification accuracy on different SNRs.

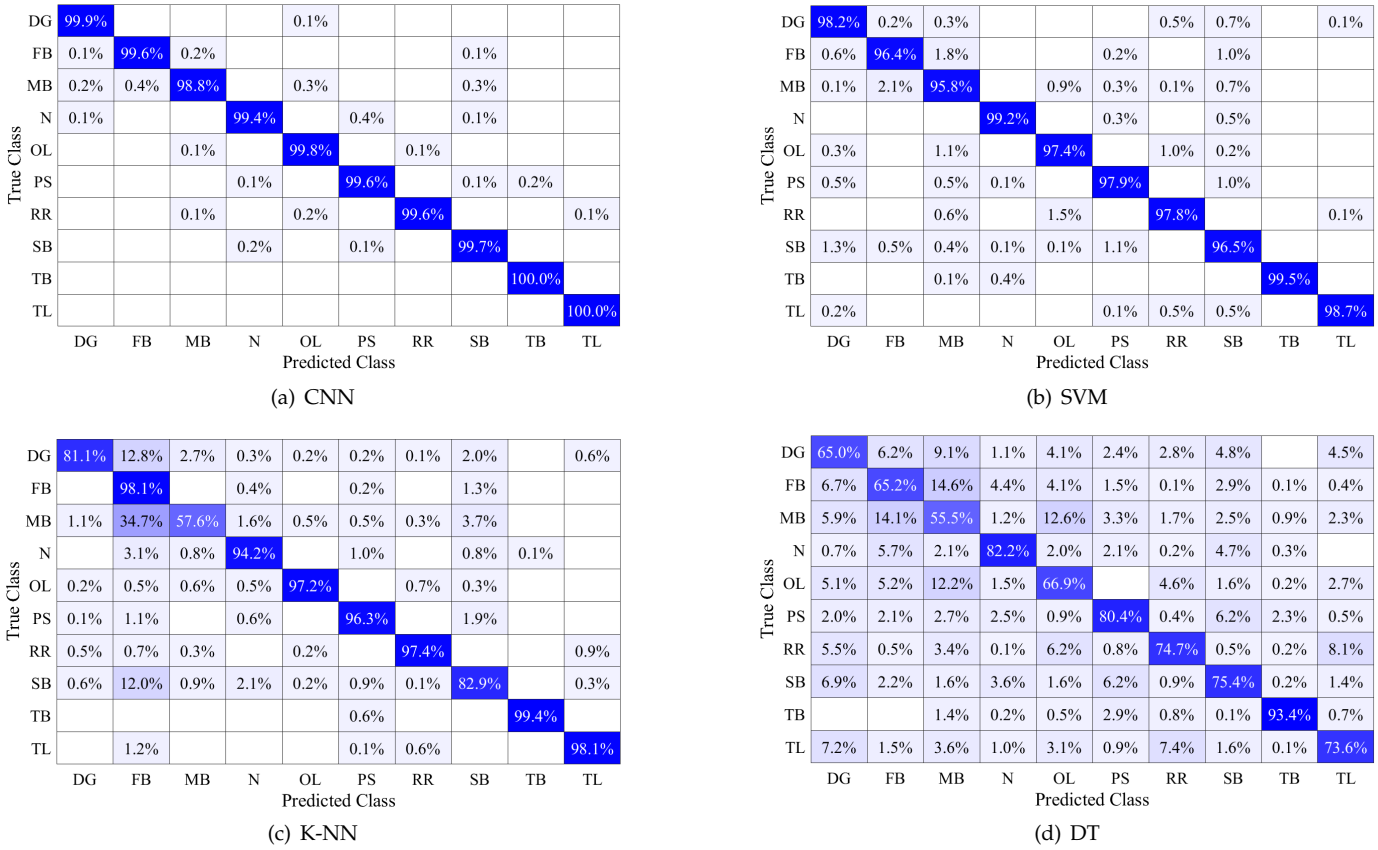
target classification in underwater acoustics.

The second experiment is conducted on the input dataset at different Signal-to-Noise Ratio (SNR): 10 dB, 5 dB, 0 dB, -5 dB, and -10 dB. In this case, CNN-CWT is used to classify the acoustic signals. The performance of the CNN model is assessed using the dataset with varying levels of noise. At low levels of noise, the model demonstrated high accuracy and robustness in classifying the data. The CNN model is able to effectively filter out the noise and make accurate predictions. As the level of noise increased, the performance of the CNN model gradually declined. The presence of higher levels of noise introduced more ambiguity into the dataset, making it more challenging for the model to correctly classify the data. From the graph in Figure 7, it can be observed that at an SNR level of 0 dB, the useful signal and the noise are approximately equal in power. However, the CWT method is still able to extract prominent features from the signal. Through the CNN classifier, these features are accumulated to achieve a highly expected classification accuracy of 96.35%.

Table IV
Comparing the Proposed CNN Model with Traditional Machine Learning Techniques

Classifier	Acc.	Train. Time	Avg. Pred. Time
CNN	99.64%	66m 54s	33.6±5.3 ms
SVM	97.74%	11m 47s	41.5±4.7 ms
K-NN	90.23%	16m 25s	432±25 ms
DT	73.23%	9m 04s	32.2±4.5 ms

The evaluation of the CNN model with the dataset at different levels of noise indicates that the model's performance is highly dependent on the noise level. This highlights the importance of noise reduction techniques or alternative models when dealing with datasets with high levels of noise. The last experiment aims to evaluate the performance of three popular machine learning methods, namely SVM, K-NN and DT, when using the same input data. Here, the CWT data is selected due to its effectiveness. The SVM employed a one-versus-one (1-vs-1) approach by constructing 45 binary training classifiers, where each target class is classified against each of the remaining 9 classes. For the K-NN, the prediction is performed using the three nearest neighbors, and the Minkowski distance metric is applied as the distance criterion. Lastly, the DT is implemented with a maximum depth of 2000 for the decision tree. The comparative results are presented in Table IV. The results in Table IV show that the SVM classifier achieved an accuracy of 97.74%. For the K-NN algorithm, experiment is conducted with different values of k ($k = 1, 3, 5, 7$), and the best result is obtained with an accuracy of 90.23% for $k = 3$. On the other hand, the accuracy of the DT classifier do not show significant improvement when changing the number of splits (83.18%). This can be explained by the limitations of the DT algorithm when applied to complex datasets like audio. CNN outperforms SVM by approximately 2%, but the training time is six times longer. This does not have much impact on real-time responsive applications because the CNN model predicts acoustic targets more fastly than the SVM model, around 8ms. Overall, based on the evaluation criterion of accuracy in Table IV, the model using the CNN classifier outperformed SVM, K-NN and DT in terms of classification accuracy. The confusion matrix of the SVM classifier in Figure 8(b) represents a high concentration of mis-



(a) CNN (b) SVM

(c) K-NN (d) DT

Figure 8. Confusion matrix of different classifiers.

predictions in certain categories, such as MusselBoat (4.2%), FishBoat (3.6%), SailBoat (3.5%) and OceanLiner (2.6%). Specifically, there is a significant amount of confusion between the pair of categories: (MusselBoat and FishBoat with 3.9%), (OceanLiner and RORO with 2.5%), (MusselBoat and OceanLiner with 2%). As for the confusion matrix of the DT classifier in Figure 8(d), it shows that the predicted errors are evenly distributed among all categories.

4 Conclusion

The article presented a study on the classification problem for 10 different target classes based on emitted acoustic signals in a underwater environment. The experiments demonstrated that the combination of the continuous wavelet transform (CWT) signal processing method and the convolutional neural network (CNN) classifier achieved the highest effectiveness (99.64%). Training the neural network required a large dataset to capture feature close to reality. However, some targets had a limited number of records (such as Trawler-1, Tugboat-2, PilotShip-2) resulting in similar patterns when divided, leading to overfitting problem. Therefore, in the future, it may be necessary to supplement the dataset to optimize the training model. Furthermore, the proposed model did not consider factors such as speed and direction of target movement, which are crucial for predicting type and estimating target parameters. Hence, it is possible to integrate the problem of determining information into the classification problem.

Acknowledgment

This work is supported by the national project under grant number ĐTĐLCN.69/23-C.

References

[1] R. J. Urick, *Principles of Underwater Sound 3rd Edition*. Peninsula Pub, 1996.

[2] S. Stergios, *Sonar Systems: Advanced Signal Processing Handbook*. InTech, dec 2000.

[3] B. Jason, *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*. Machine Learning Mastery, 2016.

[4] J. Liu, Y. He, Z. Liu, and Y. Xiong, "Underwater target recognition based on line spectrum and support vector machine," in *Proceedings of the 2014 International Conference on Mechatronics, Control and Electronic Engineering*. Atlantis Press, 2014.

[5] Y. Tong, X. Zhang, and Y. Ge, "Classification and recognition of underwater target based on MFCC feature extraction," in *Proceedings of the 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. IEEE, aug 2020.

[6] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, mar 2021.

[7] Z. Wei, Y. Ju, and M. Song, "A method of underwater acoustic signal classification based on deep neural network," in *Proceedings of the 2018 5th International*

Conference on Information Science and Control Engineering (ICISCE). IEEE, 2018, pp. 46–50.

[8] G. Jin, F. Liu, H. Wu, and Q. Song, "Deep learning based framework for expansion, recognition and classification of underwater acoustic signal," *Journal of Experimental and Theoretical Artifical Intelligence*, vol. 32, pp. 1–14, jul 2019.

[9] V.-S. Doan, T. Huynh-The, and D.-S. Kim, "Underwater acoustic target classification based on dense convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[10] K. Nikolai, *Sonar Signal Processing: Sonar Systems*. CRC Press, sep 2011.

[11] V. K. Ingle and J. G. Proakis, *Digital Signal Processing Using MATLAB 3rd Edition*. CL Engineering, jan 2011.

[12] Q. Li, *Digital Sonar Design in Underwater Acoustics*. Principles and Applications, jan 2012.

[13] S. Nagarajaiah and N. Varadarajan, "Short time fourier transform algorithm for wind response control of buildings with variable stiffness tmd," *Engineering Structures*, vol. 27, pp. 431–441, feb 2005.

[14] M. Sifuzzaman, M. Islam, and M. Ali, "Application of wavelet transform and its advantages compared to fourier transform," *Journal of Physical Sciences*, vol. 13, pp. 121–134, 2009.

[15] D. Santos-Domínguez, S. Torres-Guijarro, A. Cardenal-López, and A. Pena-Gimenez, "ShipsEar: An underwater vessel noise database," *Applied Acoustics*, vol. 113, pp. 64–69, dec 2016.

Ngoc Anh Tu Nguyen received his Engineer degree in radar and navigation from Le Quy Don Technical University in 2016. He is now a student in Master program in Le Quy Don Technical University. His current research interests are radar and sonar systems, signal processing and deep learning. Email:nguyentuhvhq@gmail.com

Van Sang DOAN received his M.Sc. and Ph.D. degrees in electronic systems and devices from Faculty of Military Technology, University of Defence, Brno, Czech Republic, in 2013 and 2016, respectively. He was awarded three Honor medals by the Faculty of Military Technology, the University of Defence in 2011, 2013, and 2016, respectively. From 2019 to 2020, he was a postdoctoral research fellow at ICT Convergence Research Center, Kumoh National Institute of Technology, South Korea. He is currently a lecturer at Faculty of Communication and Radar, Vietnam Naval Academy, Nha Trang City, Khanh Hoa Province, Vietnam. His current research interests include radar, sonar and communication systems, signal processing, and deep learning. Email: doansang.g1@gmail.com