Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.089269

ARTICLE

# Deep Learning-Based Underwater Propeller Recognition Using Time-Frequency Representations of Acoustic Signals

**Erkan Caner Ozkat***

Department of Mechanical Engineering, Faculty of Engineering and Architecture, Recep Tayyip Erdogan University, Rize, Türkiye

*Corresponding Author: Erkan Caner Ozkat. Email: erkancaner.ozkat@erdogan.edu.tr

Received: 17 July 2026; Accepted: 21 October 2026; Published: 17 November 2026

**ABSTRACT:** Underwater propeller recognition supports maritime surveillance, defense, and environmental noise monitoring. Published classifiers report near-perfect accuracy on the public propeller blade-count dataset, yet all rely on segment-level cross-validation. Segments cut from one continuous recording then appear in both training and test folds, which leaks recording-specific information. This study presents a leakage-aware benchmark of deep learning for propeller blade-count recognition. Seven classifiers are re-evaluated under two protocols. The first is the established stratified segment-level protocol. The second is a blocked recording-level protocol with purging, which tests extrapolation across time. The models span four ImageNet transfer networks, a from-scratch multi-scale convolutional neural network (CNN), a raw-waveform one-dimensional CNN, and a feature-based support vector machine. Model selection uses inner validation only, and all comparisons carry paired statistical tests. Three findings emerge. First, no model reaches the published 99.80% reference level once model selection avoids the test fold. Second, the protocol changes stability far more than mean accuracy. Every spectrogram transfer model falls to 72%–79% on the first temporal block, while mean accuracy shifts by only two points. Third, the one-dimensional CNN with 0.46M parameters is the strongest and most robust model. It reaches $99.13 \pm 0.73\%$ under the segment-level protocol and $98.27 \pm 2.89\%$ under the blocked protocol. Gradient-weighted class activation mapping confirms that the networks read blade-pass harmonics below 1 kHz. Ablations show that log compression is essential, worth 19.8 percentage points, while augmentation gives no significant gain. The evaluation protocol, not the architecture, governs the reported accuracy on this dataset.

**KEYWORDS:** Underwater acoustics; propeller recognition; deep learning; data leakage; cross-validation

## 1 Introduction

Underwater propeller acoustic signatures support naval surveillance, port security, marine traffic management, and environmental noise monitoring [1,2]. Acoustic sensing operates passively at long range and requires no physical contact with the target. These properties make it well suited to military and civilian maritime monitoring [3,4]. The underlying physics are well established. A propeller's acoustic signature reflects its geometry, blade count, and shaft speed [5,6]. Radiated energy concentrates at blade-pass frequencies and their harmonics. Tip and hub vortex cavitation produce a broadband spectral floor [7,8]. Kimmerl et al. [7] used implicit large-eddy simulation to show that cavitation dynamics dominate the acoustic source. A companion study [8] confirmed via the Ffowcs Williams–Hawkings analogy that turbulence resolution governs near-field pressure prediction. Zhu et al. [9] analysed a controllable-pitch vessel using spectral density and coherence methods. They found that the 50% pitch setting maximises tonal energy. Each

Copyright © 2026 The Author. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

propeller configuration therefore produces a consistent and distinctive spectral fingerprint. The problem is also pressing from an environmental standpoint. Ambient ocean noise levels have risen roughly 3.3 dB per decade since 1950 [10]. Commercial shipping is identified as the primary driver of this increase. Regulators are increasingly formalising vessel noise limits [7]. Propeller noise dominates vessel acoustics above 27 km/h; machinery noise takes over at lower speeds [10]. The physical distinctiveness of propeller signatures, together with these regulatory drivers, makes automated recognition well motivated.

Several classification approaches build on this physical foundation. Aslam et al. [11] surveyed over 250 papers on marine vessel and fish sound classification. Their review illustrates the breadth of machine learning and deep learning applications in the field. On the signal processing side, Filho et al. [12] modified the detection of envelope modulation on noise (DEMON) analysis. They replaced its short-time Fourier transform (STFT) with a Hilbert–Huang Transform (HHT). This was combined with support vector machine (SVM) noise rejection in their HHT with intelligent noise reduction (HHT-INR) method. The approach gained an average of 87.9 dB signal-to-noise ratio (SNR) through targeted empirical mode decomposition. Wang et al. [13] introduced the Adaptive Generalized Warblet Transform (AGWT) to extract micro-Doppler features from low-frequency line spectra. Their results show that wave-induced modulation aids long-range target discrimination. On the classification side, Khishe and Mosavi [14] trained a neural network with the Chimp Optimization Algorithm. González-Hernández et al. [15] applied octave-band analysis to marine mammal recognition. Khishe and Safari [16] used a dragonfly-algorithm-optimised multi-layer perceptron (MLP) for sonar targets. Tuncer et al. [17] proposed the dynamic center mirror local binary pattern (DCMLBP) descriptor for ambient sound identification. The current state of the art on the propeller dataset is the Data Encryption Standard (DES) pattern method, DES-Pat, of Yaman et al. [18]. DES-Pat extracts 512 features via DES substitution-box (S-Box) transformations and reduces them to 150 dimensions with Neighbourhood Component Analysis (NCA). Classification is performed with SVM, reaching 99.80% accuracy, ahead of the k-nearest neighbours (KNN) and decision tree classifiers tested in the same study [18]. These results confirm that propeller sounds carry enough class-specific information for high-accuracy recognition. Table 1 summarises the most closely related studies, the problems they address, their methods, and their main findings, and positions the present work against them.

**Table 1:** Summary of closely related studies on underwater acoustic classification.

| Study | Problem | Method/Tool | Data | Key Finding |
|---|---|---|---|---|
| Yaman et al. [18] | Propeller blade-count recognition | DES S-Box features, NCA, SVM (MATLAB) | Propeller dataset | 99.80% under segment-level 10-fold cross-validation |
| Khishe and Mosavi [14] | Sonar dataset classification | MLP trained by Chimp Optimization | Sonar targets | Metaheuristic training improves MLP accuracy |
| Tuncer et al. [17] | Ambient sound identification | DCMLBP local binary patterns | Acoustic recordings | Handcrafted descriptor with shallow classifiers |
| Filho et al. [12] | Propulsion signature extraction | HHT-based DEMON with SVM noise rejection | Passive sonar | Average 87.9 dB SNR gain over STFT DEMON |

(Continued)

**Table 1 (continued)**

| Study | Problem | Method/Tool | Data | Key Finding |
|---|---|---|---|---|
| Wang et al. [13] | Low-frequency line spectra | Adaptive Generalized Warblet Transform | Sea-trial data | Micro-Doppler features aid long-range discrimination |
| Yang et al. [19] | Ship type recognition | SE-ResNet-10, 3D MFCC | ShipsEar | Channel attention boosts spectro-temporal features |
| Liu et al. [20] | Surface ship recognition | Physics-informed fuzzy network, WGAN augmentation | DeepShip | 91.22% precision under open-sea conditions |
| Karabacak [21] | Tool wear stage estimation | Transfer learning CNNs on STFT images | Machining signals | EfficientNet-B0 most parameter-efficient backbone |
| This work | Propeller blade-count recognition | EfficientNet-B0 on log-mel-spectrograms, leakage-aware benchmark | Propeller dataset | Evaluation protocol governs reported accuracy |

All these methods share a fundamental weakness. Hand-crafted representations encode statistical signal properties but miss the spectro-temporal structure of propeller acoustics. This applies whether features come from cipher transformations, spectral envelopes, or fixed filterbanks. Fixed pipelines need careful manual tuning and cannot adapt when conditions shift. The underwater channel further compounds the problem. Frequency-dependent absorption, multipath propagation, variable water-column profiles, Doppler shifts, and low SNR distort the feature space [11]. Static representations cannot capture these distortions. SVM classifiers are particularly ill-suited here. They lack the inductive bias to learn context-dependent spectro-temporal patterns. Prior work in underwater acoustics either relies on hand-crafted features [14,15] or applies neural networks directly to raw waveforms [16]. Spectrogram-based deep learning is well established on ship-radiated-noise corpora such as ShipsEar and DeepShip [19,20]. On the propeller blade-count dataset, however, published evaluations rely on a segment-level cross-validation protocol, and no leakage-aware, statistically tested benchmark exists. A second open question follows from the first. Does the reported accuracy reflect blade-count physics or recording-specific artifacts? Only a recording-level protocol and a feature-attribution analysis can answer this.

Evidence from related fields indicates that spectrogram-based convolutional neural networks (CNNs) are well suited to this problem. CNNs trained on mel-spectrograms and STFT images perform well across audio classification tasks [22,23]. These tasks range from speech recognition to environmental sound identification. Residual networks [24], dense networks [25], mobile architectures [26], and compound-scaled networks [27] are routinely fine-tuned from ImageNet weights. Adam [28] is the standard optimiser for this purpose. SpecAugment [29] masking further improves robustness to missing spectral regions. Transformer-based audio classifiers have also emerged [30]. Underwater-specific designs include a multiscale residual network [31] and a squeeze-and-excitation ResNet-10 with 3D mel-frequency cepstral coefficient (MFCC)

features on the ShipsEar dataset [19]. Most recently, physics-informed architectures have been proposed for open-sea ship-radiated noise. Liu et al. [20] introduced MmPiFNN, a multi-mode physics-informed fuzzy neural network that fuses time, time-frequency, and spatial features under a physics-based residual loss. It reaches 91.22% precision on DeepShip. Wang et al. [32] proposed PC-LNN, a physics-constrained liquid neural network for multi-scale ship-radiated noise recognition. These works target uncontrolled open-sea recordings with many latent factors. The present study is complementary. It isolates a single physical factor, the blade count, under controlled conditions. It also quantifies how the evaluation protocol itself changes the reported performance. Karabacak [21] benchmarked transfer learning backbones on STFT spectrograms from a machining monitoring task. EfficientNet-B0 [27] proved the most parameter-efficient among ResNet-50, DenseNet-121, and MobileNetV3 in that study.

This work asks two questions. The first is whether CNN transfer learning on time-frequency representations can replace hand-crafted feature pipelines for propeller recognition. The second is how much of the near-perfect accuracy reported on this dataset survives when the evaluation protocol prevents segment-level leakage. EfficientNet-B0 [27] is fine-tuned on log-mel-spectrograms and compared against the DES-Pat reference [18] and several deep learning baselines under both protocols. The study is a benchmarking and evaluation-protocol contribution; no claim of architectural novelty is made. The main contributions are as follows:

(i) A leakage-aware benchmark of propeller blade-count recognition is presented. Every model is evaluated under the established stratified segment-level protocol and under a blocked recording-level protocol with purging. The comparison quantifies, for the first time on this dataset, how strongly the protocol choice shapes the reported accuracy.
(ii) A systematic comparison of transfer learning architectures and lightweight networks is conducted with paired statistical tests, replacing single-number rankings.
(iii) A physical-interpretation analysis links the network's gradient-weighted class activation mapping (Grad-CAM) saliency to blade-pass-frequency harmonic structure and examines class separability in the learned embedding space.
(iv) A lightweight one-dimensional CNN (0.46M parameters) operating on raw waveforms is evaluated as an edge-oriented alternative, with measured desktop latency and explicitly qualified deployment claims.

The remainder of the paper is structured as follows. Section 2 describes the dataset and preprocessing pipeline. Section 3 presents the methodology. Section 4 details the experimental setup. Section 5 reports the results and comparisons. Section 6 discusses the findings. Section 7 concludes the paper and outlines future work.

## 2 Materials

### *2.1 Dataset Description*

The dataset was collected by Yaman et al. [18]. It is publicly available at https://github.com/orhanyaman/Propeller. The dataset contains underwater acoustic recordings from five propellers. The propellers differ mainly in blade count: two, three, four, five, and six blades. Hub length, blade root, and maximum thickness vary between classes as well.

Each propeller was modelled in SolidWorks and 3D-printed. Geometric specifications are listed in Table 2. Recordings were made in a 200 cm × 100 cm × 100 cm water tank. A 12 V DC motor drove each propeller during the recordings. A waterproofed microphone was placed at 100 cm standoff distance. Audio was captured using a voice recorder app on an LG G4 smartphone. The recordings were saved in AAC (m4a) format at 320 kbps and 44 kHz. Each propeller was recorded for roughly 300 s. The total duration is approximately 1500 s across the five classes.

**Table 2:** Geometric dimensions of the five propellers (mm) [18].

| Parameter | Two Blades | Three Blades | Four Blades | Five Blades | Six Blades |
|---|---|---|---|---|---|
| Diameter | 30 | 30 | 30 | 30 | 27 |
| Radius | 15 | 15 | 15 | 15 | 15 |
| Blade Number | 2 | 3 | 4 | 5 | 6 |
| Hub | 7 | 7 | 7 | 7 | 7 |
| Hub Length | 7.5 | 7.5 | 11 | 8 | 8 |
| Blade Root | 15 | 15 | 27.5 | 16 | 20 |
| Maximum Thickness | 2 | 2 | 2.5 | 2 | 2.5 |
| Maximum Camber | 15 | 15 | 15 | 15 | 15 |

### *2.2 Audio Preprocessing Pipeline*

The preprocessing pipeline consists of five sequential steps with fixed parameters.

1. **Decoding and resampling.** The raw m4a files are decoded to pulse-code modulation (PCM) and resampled to 22,050 Hz with a polyphase filter.
2. **Segmentation.** The decoded recordings last 323, 313, 375, 348, and 340 s for the two- to six-blade classes. The first 300 s of each recording are used, which keeps the classes balanced and matches the duration stated in the source publication [18]. Each recording is cut into non-overlapping 1-s windows, yielding 1500 segments with 300 segments per class.
3. **Normalisation.** Each segment is normalised to zero mean and unit variance.
4. **Time-frequency transform.** Mel-spectrograms are computed with 128 mel bands, a 2048-point fast Fourier transform (FFT), a Hann window, and a 512-sample hop, giving a 128 × 45 matrix per segment. Log compression follows as $S_{log} = \log(S_{mel} + \epsilon)$ with $\epsilon = 10^{-6}$.
5. **Resizing.** The log-mel-spectrograms are min–max normalised, bilinearly resized to 224 × 224, and replicated across three channels to match the pretrained CNN input format. ImageNet channel statistics are then applied.

Fig. 1 shows representative log-mel-spectrograms for each class. Tonal energy below approximately 2 kHz corresponds to blade-pass frequency (BPF) harmonics. The harmonics become more closely spaced and denser as the blade count rises from two to six. This follows the linear relationship BPF = $N \times \Omega/60$ between blade count $N$ and shaft speed $\Omega$ in rpm. The broadband energy at higher frequencies reflects tip vortex cavitation noise, whose intensity grows with blade count. These class-specific signatures are the discriminative structure the classifier exploits.

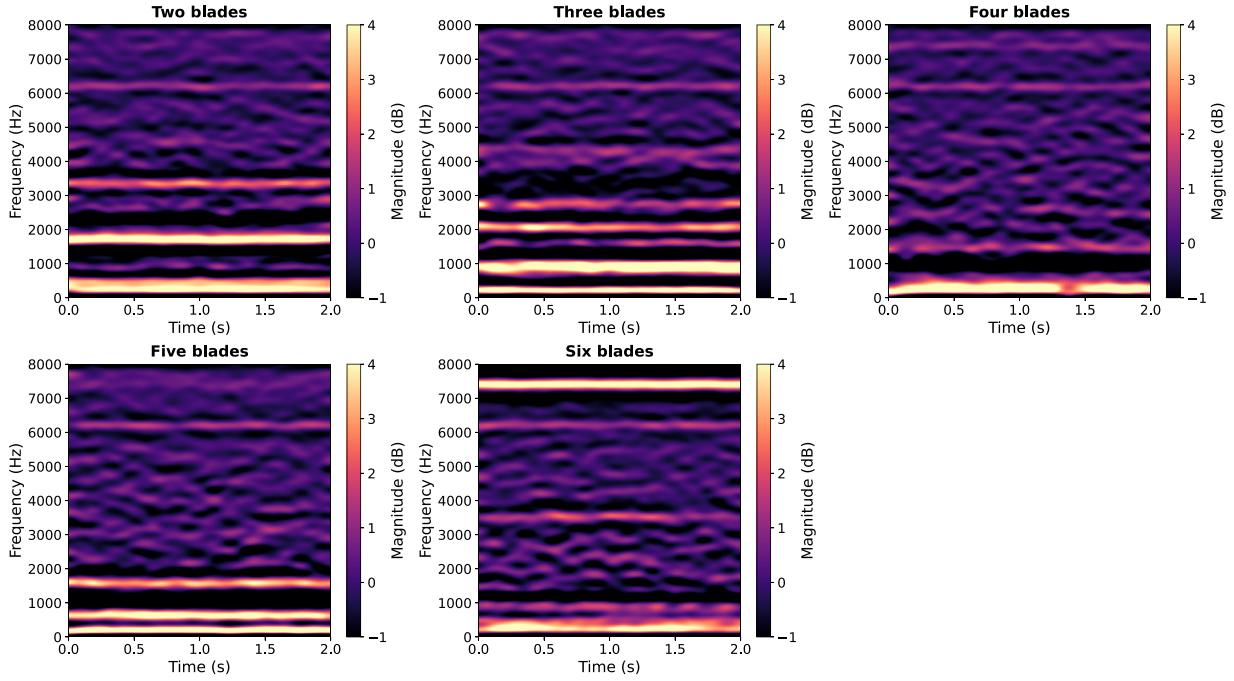



**Figure 1:** Log-mel-spectrograms of one representative acoustic segment per propeller class.

### *2.3 Data Augmentation*

Four augmentation operations were applied online during training. Test and validation segments were never augmented. Each operation is applied independently with probability 0.5 per sample per epoch. A sample can therefore receive any of the $2^4 = 16$ operation combinations. The operations, their parameter ranges, and their physical rationale are as follows.

- **Time stretching** (waveform domain). A random factor is drawn uniformly from $[0.8, 1.2]$ and the segment is resampled to its original length. This mimics shaft-speed variation, which shifts the blade-pass frequency and its harmonic spacing.
- **Additive Gaussian noise** (waveform domain). White noise is added at a signal-to-noise ratio drawn uniformly from $[20, 40]$ dB. This mimics broadband ambient noise.
- **Circular time shifting** (waveform domain). The segment is rolled by an offset drawn uniformly from $[-0.1, 0.1]$ s. This enforces invariance to the segment start position.
- **SpecAugment** [29] (spectrogram domain, spectrogram models only). One frequency mask of up to 20 mel bins and one time mask of up to 10 frames are zeroed. This encourages robustness to missing spectral regions.

## 3 Proposed Methodology

Fig. 2 shows the proposed framework. Two parallel pipelines cover the accuracy–efficiency spectrum. The primary pipeline converts acoustic signals to log-mel-spectrograms and fine-tunes EfficientNet-B0. The secondary pipeline offers a lightweight alternative for edge deployment. It uses either the multi-scale feature fusion CNN (MSFF-CNN) on spectrograms or the one-dimensional CNN (1D-CNN) on raw waveforms.

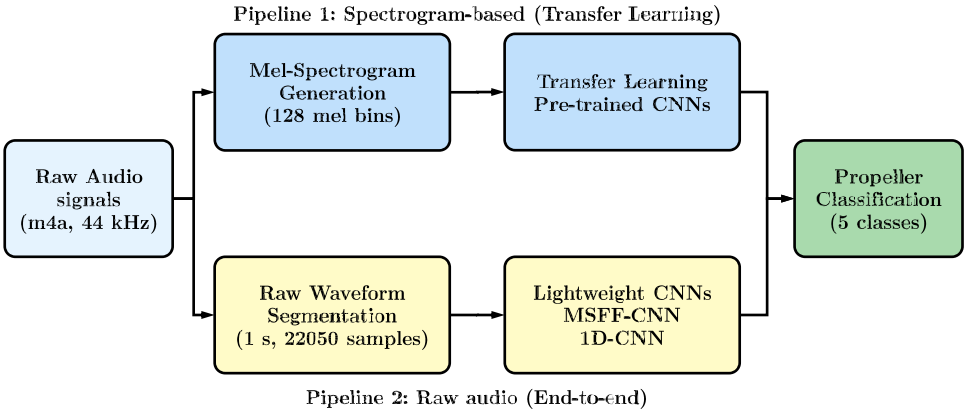



**Figure 2:** Overview of the proposed two-pipeline framework.

### *3.1 Proposed Model: EfficientNet-B0 with Transfer Learning*

EfficientNet [27] uses compound scaling to uniformly scale network width, depth, and resolution. A compound coefficient $\phi$ controls all three dimensions:

$$\text{depth: } d = \alpha^{\phi}, \quad \text{width: } w = \beta^{\phi}, \quad \text{resolution: } r = \gamma^{\phi} \tag{1}$$

subject to $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$. EfficientNet-B0 corresponds to $\phi = 0$. With the five-class head used here it holds 4.34M parameters, yet it matches or exceeds much larger models in accuracy.

Three problem-specific reasons motivated this choice. First, compound scaling jointly optimises width, depth, and resolution. This is a natural fit for log-mel-spectrogram inputs where the frequency and time axes both carry physically meaningful information. Scaling only one dimension risks saturating capacity along one axis while underusing the other. Second, EfficientNet-B0 employs Mobile Inverted Bottleneck Convolution (MBConv) blocks. These blocks pair depthwise separable convolutions with squeeze-and-excitation channel attention. This lets the network recalibrate the importance of individual mel bands. The mechanism is directly relevant when blade-pass harmonics occupy class-specific frequency regions. Third, 4.34M parameters is well matched to a training set of roughly 1200 fitted samples per fold. Heavier models such as ResNet-50 (24.03M) and DenseNet-121 (7.22M) add computational cost without significant accuracy gains, as Section 5 confirms. Karabacak [21] reached the same conclusion in a spectrogram-based machinery monitoring benchmark.

The ImageNet-pretrained backbone was adapted by replacing its classification head. The new head consisted of five layers stacked in sequence. The first layer was Global Average Pooling (GAP). It was followed by Dropout with rate 0.3. A Dense (256) layer with rectified linear unit (ReLU) activation was then applied. A second Dropout (0.3) layer was added next. The final layer was a Dense (5, Softmax) classifier. Training proceeded in two phases:

1. **Phase 1, warm-up (5 epochs)**: backbone frozen, only the new head trained at $lr = 10^{-3}$.
2. **Phase 2, fine-tuning (up to 45 epochs)**: all layers unfrozen, $lr = 10^{-4}$ to avoid catastrophic forgetting.

### *3.2 Benchmark Models*

EfficientNet-B0 was evaluated against five benchmark architectures spanning the accuracy–efficiency spectrum. All parameter counts below refer to the five-class configurations actually trained. ResNet-50 [24] has 50 layers with residual skip connections and 24.03M parameters; it serves as the heavyweight

baseline. MobileNetV3 [26] targets mobile deployment with 3.22M parameters, using depthwise separable convolutions and squeeze-and-excitation modules. DenseNet-121 [25] connects all preceding layers to each subsequent layer with 7.22M parameters. The custom MSFF-CNN is a lightweight multi-scale feature fusion network with 0.37M parameters. It stacks three inception-style blocks with parallel $1 \times 1$, $3 \times 3$, and $5 \times 5$ branches and squeeze-and-excitation attention. It is trained from scratch on the same $224 \times 224$ log-mel images as the transfer models, with no pretraining. Finally, the 1D-CNN (0.46M parameters) operates directly on raw audio waveforms. It uses four convolutional blocks with progressively decreasing kernel sizes (64, 32, 16, 8), each with batch normalisation and max pooling. Global average pooling aggregates the final block output before the dense classifier (Fig. 3, Table 3). A feature-based baseline completes the comparison. It concatenates MFCCs with their first and second derivatives, mel-band statistics, spectral centroid, bandwidth, roll-off, flatness, and contrast, chroma, zero-crossing rate, and root-mean-square (RMS) energy. A radial-basis-function SVM with $C = 100$ classifies the resulting vector.

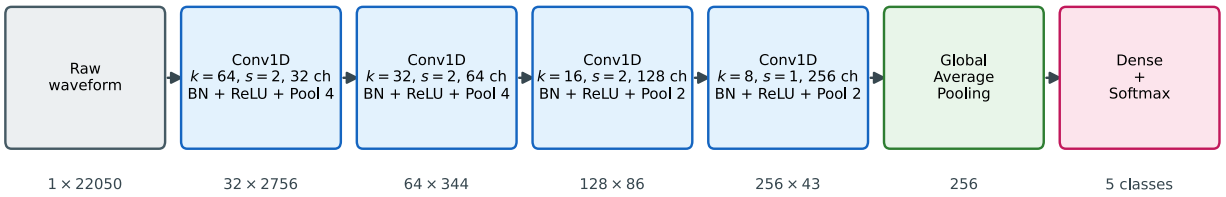



**Figure 3:** Architecture of the 1D-CNN benchmark for raw audio classification.

**Table 3:** Architecture of the 1D-CNN benchmark.

| Layer | Filters | Kernel | Stride | Pool | Output |
|---|---|---|---|---|---|
| Input | – | – | – | – | 1 × 22,050 |
| Conv1D Block 1 | 32 | 64 | 2 | 4 | 32 × 2756 |
| Conv1D Block 2 | 64 | 32 | 2 | 4 | 64 × 344 |
| Conv1D Block 3 | 128 | 16 | 2 | 2 | 128 × 86 |
| Conv1D Block 4 | 256 | 8 | 1 | 2 | 256 × 43 |
| Global Average Pooling | – | – | – | – | 256 |
| Dense + Softmax | 5 | – | – | – | 5 |

### *3.3 Training Procedure*

All models used the Adam optimiser [28]. The batch size was 32 with up to 50 epochs. Early stopping was applied with patience 10. The loss function was categorical cross-entropy. Learning rate scheduling used ReduceLROnPlateau with factor 0.5 and patience 5. Model selection never touches the test fold. An inner validation set holding 10% of the training portion drives early stopping, checkpoint selection, and the learning-rate schedule. In the stratified protocol this set is drawn by stratified random sampling. In the blocked protocol it is the temporally last 10% of each class's training segments, which keeps the split contiguous. Each test fold is evaluated exactly once, after training completes. Algorithm 1 summarises the full two-phase transfer-learning workflow.

**Algorithm 1:** Two-phase transfer learning with inner-validation checkpointing

**Require:** training segments $\mathcal{D}_{tr}$, test segments $\mathcal{D}_{te}$, fold seed $s$
1: Split $\mathcal{D}_{tr}$ into fit set $\mathcal{D}_{fit}$ (90%) and inner validation set $\mathcal{D}_{val}$ (10%)
2: Convert every segment to a log-mel-spectrogram and resize to $224 \times 224 \times 3$
3: Initialise EfficientNet-B0 with ImageNet-1K weights; replace the head with GAP–Dropout (0.3)–Dense (256, ReLU)–Dropout (0.3)–Dense (5, Softmax)
4: **Phase 1 (warm-up):** freeze the backbone; train the head for 5 epochs with Adam, $lr = 10^{-3}$
5: **Phase 2 ( fine-tuning):** unfreeze all layers; continue with Adam, $lr = 10^{-4}$
6: **for** each epoch until epoch 50 or patience 10 exceeded **do**
7: Apply augmentation to $\mathcal{D}_{fit}$ with probability 0.5 per operation per sample
8: Train one epoch; evaluate accuracy on $\mathcal{D}_{val}$
9: **if** validation accuracy improved **then** store checkpoint; reset patience
10: **else** decay $lr$ by 0.5 after 5 stagnant epochs; increment patience
11: **end for**
12: Restore the best stored checkpoint
13: Evaluate once on $\mathcal{D}_{te}$ and report accuracy (ACC), unweighted average precision (UAP), unweighted average recall (UAR), and F1

## 4 Experimental Setup

All models were evaluated under two complementary 10-fold cross-validation protocols, and every result table reports both.

**Protocol A: stratified segment-level cross-validation.** This is the protocol of Yaman et al. [18] and of the original version of this study. The 1500 segments are shuffled and split into ten stratified folds. Each fold allocates 150 test samples (30 per class) and 1350 training samples. Because each class stems from one continuous recording, segments that were seconds apart in the same recording can appear on both sides of the split. The protocol therefore measures within-recording interpolation and enables direct comparison with the published DES-Pat results.

**Protocol B: blocked recording-level cross-validation.** Each class is represented by a single continuous recording, so a strict recording-level split would remove the class entirely. The closest leakage-aware alternative divides each recording into ten contiguous temporal blocks of 30 s. Fold $k$ tests block $k$ of every class and trains on the remaining blocks. One segment on each side of the test block is purged from the training set to suppress boundary correlation. This protocol measures extrapolation across time within the same recording session, which is the sternest test the single-recording-per-class design permits.

In both protocols, results are reported as mean ± standard deviation over the ten folds, together with the fold minimum. Per-fold values are retained for the paired statistical tests of Section 5.

Four metrics were reported in this study. These are accuracy (ACC), unweighted average precision (UAP), unweighted average recall (UAR), and the macro-averaged F1-score. ACC measures the fraction of correctly classified samples. UAP and UAR are the macro-averages of precision and recall over all five classes. The macro-averaged F1-score is the unweighted mean of the per-class F1-scores. Reporting UAP and UAR alongside ACC guards against class-imbalance artefacts and enables direct comparison with published multi-class benchmarks. Let $TP_c$, $FP_c$, and $FN_c$ denote the true positives, false positives, and false negatives of class $c$. With $N$ test samples, the metrics are defined as

$$\text{ACC} = \frac{1}{N}\sum_{c=1}^{5} TP_c \tag{2}$$

$$\text{UAP} = \frac{1}{5}\sum_{c=1}^{5} \frac{TP_c}{TP_c + FP_c} \tag{3}$$

$$\text{UAR} = \frac{1}{5}\sum_{c=1}^{5} \frac{TP_c}{TP_c + FN_c} \tag{4}$$

$$\text{F1} = \frac{1}{5}\sum_{c=1}^{5} \frac{2P_c R_c}{P_c + R_c} \tag{5}$$

where $P_c$ and $R_c$ are the precision and recall of class $c$.

The primary reference is DES-Pat [18]. DES-Pat reaches 99.80% accuracy with Quadratic/Cubic SVM on 150 NCA-reduced features. Additional shallow baselines from the same work are Fine Decision Tree (99.00%) and Fine KNN (99.60%). These references were chosen to position the proposed framework against the current state of the art on the same dataset.

### *4.1 Hardware and Software Environment*

All experiments were implemented in Python 3.12.3 with PyTorch 2.6.0, torchvision 0.21.0, librosa 1.0.0, SciPy 1.18.1, NumPy 2.5.2, and scikit-learn 1.9.0. Pretrained weights were obtained from the torchvision ImageNet-1K model zoo. The transfer-learning models were trained on an NVIDIA GeForce GTX 1050 GPU with 4 GB memory using the CUDA 12.4 build of PyTorch. The 1D-CNN and the feature-based baselines were trained on an Intel Core i7-7700HQ CPU (8 logical cores) with 32 GB RAM. All training and inference times reported in this paper were measured on this hardware. Random seeds were fixed to 42 plus the fold index for every fold.

The initialisation of the added layers is specified as follows. Each convolutional backbone starts from the torchvision ImageNet-1K weights. The classification layers that replace the original ImageNet classifier are newly created and therefore carry no pretrained values. They use the PyTorch default scheme for linear layers. Weights are drawn from a Kaiming-uniform distribution in fan-in mode with the negative slope parameter $a = \sqrt{5}$. Biases are drawn from a uniform distribution bounded by $\pm 1/\sqrt{fan_in}$. The MSFF-CNN and the 1D-CNN are trained from scratch, so every layer in those two models uses the same default scheme. The per-fold seed governs the fold partition, the inner-validation split, the ordering of the training batches, and the sampling of the augmentation operations. It is set at the start of training, which is after the network object is constructed. The draw of the initial weights therefore comes from the process-level generator rather than from the fold seed. Repeated training therefore reproduces the reported fold statistics but need not reproduce a given fold to the last decimal. The released checkpoints remove this ambiguity for the evaluation stage, because they reproduce the reported test predictions exactly.

## 5 Results

This section reports the classification performance, the ablations, the computational cost, and the physical interpretation of the learned features.

### *5.1 Classification Performance*

Table 4 reports the proposed model performance under both evaluation protocols. All values come from the re-executed experiments described in Section 4.

**Table 4:** Classification performance (%) of the proposed EfficientNet-B0 model with log-mel input under both 10-fold cross-validation protocols.

| Protocol | ACC | UAP | UAR | F1 | Min Fold ACC |
|---|---|---|---|---|---|
| A: stratified | 97.00 ± 1.77 | 97.09 ± 1.74 | 97.00 ± 1.77 | 97.01 ± 1.77 | 92.67 |
| B: blocked | 95.20 ± 7.92 | 96.30 ± 5.94 | 95.20 ± 7.92 | 95.15 ± 8.15 | 74.67 |

Two observations follow. First, the segment-level result is 97.00 ± 1.77%. Selecting checkpoints on the test fold instead yields 99.87%, which overstates the achievable accuracy. Here checkpoints are chosen on an inner validation split, and the test fold is evaluated once. Second, the blocked protocol leaves the mean within two percentage points but multiplies the standard deviation by four. The first and last temporal blocks of the recordings drop to 74.67% and 86.00%. Fig. 4 shows the per-fold accuracies of both protocols. Segment-level evaluation therefore overstates the temporal stability of the classifier far more than its average accuracy.

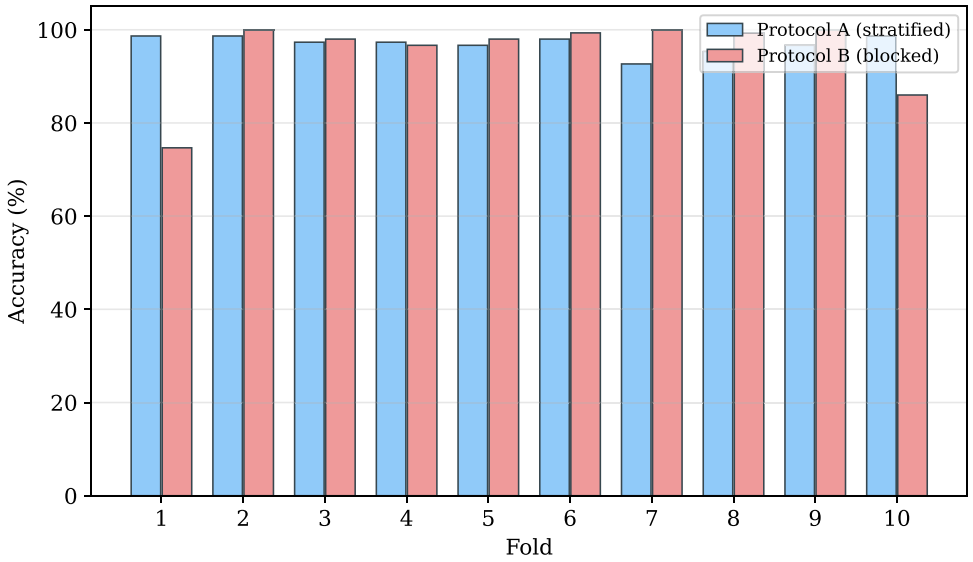



**Figure 4:** Per-fold test accuracy of the proposed model under both protocols.

Table 5 and Fig. 5 compare all methods under both protocols. The published DES-Pat results are quoted for context. They were obtained under a segment-level protocol comparable to Protocol A and were not re-executed in this study. All metrics are means over the ten folds with their standard deviations, and Min is the worst fold accuracy. Every fold is class balanced by construction, so UAR coincides with ACC throughout.

Four findings emerge. First, the 1D-CNN on raw waveforms is the strongest model under both protocols. Under Protocol A its advantage is statistically significant against EfficientNet-B0 (paired t-test $p = 0.006$, Wilcoxon $p = 0.004$) and against the feature-based SVM ($p = 0.023$). Under Protocol B it outperforms ResNet-50, DenseNet-121, and EfficientNet-B0 by the Wilcoxon test ($p < 0.05$). Its margins over MobileNetV3 ($p = 0.34$) and over the SVM are not significant. Second, the four ImageNet transfer models cluster between 97.0% and 98.0% under Protocol A and between 95.2% and 96.3% under Protocol B. No transfer model reaches the published DES-Pat level once model selection avoids the test fold. Third, the protocol changes the fold minimum far more than the mean. Every spectrogram transfer model drops to 72%–79% on the first temporal block, while the 1D-CNN never falls below 91%. Fourth, the from-scratch MSFF-CNN is the weakest and least stable model. That weakness reflects the capacity of its architecture rather

than the absence of pretraining. Section 5.2 isolates the two factors. EfficientNet-B0 trained from random initialisation matches its pretrained counterpart on this dataset. The smallest raw-waveform network remains the best overall choice on this dataset.

**Table 5:** Comparison of all re-executed methods under both protocols, with the published DES-Pat reference.

| Method | ACC (%) | UAP (%) | UAR (%) | F1 (%) | Min |
|---|---|---|---|---|---|
| *Published reference, not re-executed [18]* DES-Pat + Quadratic/Cubic SVM | 99.80 | – | – | – | – |
| *Protocol A: stratified, segment level* | | | | | |
| MultiDomain features + SVM | 98.47 ± 0.73 | 98.58 ± 0.64 | 98.47 ± 0.73 | 98.48 ± 0.72 | 97.33 |
| MSFF-CNN ( from scratch) | 93.33 ± 5.63 | 94.65 ± 4.15 | 93.33 ± 5.63 | 93.38 ± 5.61 | 82.00 |
| 1D-CNN (raw waveform) | 99.13 ± 0.73 | 99.17 ± 0.70 | 99.13 ± 0.73 | 99.13 ± 0.73 | 98.00 |
| MobileNetV3 + Log-mel | 98.00 ± 0.79 | 98.08 ± 0.76 | 98.00 ± 0.79 | 98.00 ± 0.78 | 96.67 |
| DenseNet-121 + Log-mel | 97.93 ± 1.13 | 97.99 ± 1.09 | 97.93 ± 1.13 | 97.93 ± 1.13 | 95.33 |
| ResNet-50 + Log-mel | 97.53 ± 1.37 | 97.65 ± 1.28 | 97.53 ± 1.37 | 97.54 ± 1.36 | 96.00 |
| EfficientNet-B0 + Log-mel | 97.00 ± 1.77 | 97.09 ± 1.74 | 97.00 ± 1.77 | 97.01 ± 1.77 | 92.67 |
| *Protocol B: blocked, temporal* | | | | | |
| MultiDomain features + SVM | 97.80 ± 4.77 | 98.25 ± 3.59 | 97.80 ± 4.77 | 97.85 ± 4.60 | 84.00 |
| MSFF-CNN ( from scratch) | 91.60 ± 9.93 | 93.84 ± 7.01 | 91.60 ± 9.93 | 90.99 ± 11.47 | 67.33 |
| 1D-CNN (raw waveform) | 98.27 ± 2.89 | 98.50 ± 2.45 | 98.27 ± 2.89 | 98.29 ± 2.85 | 91.33 |
| MobileNetV3 + Log-mel | 96.33 ± 6.97 | 96.77 ± 6.11 | 96.33 ± 6.97 | 96.34 ± 6.90 | 76.00 |
| DenseNet-121 + Log-mel | 96.33 ± 5.94 | 96.99 ± 4.69 | 96.33 ± 5.94 | 96.35 ± 5.92 | 79.33 |
| ResNet-50 + Log-mel | 95.67 ± 8.00 | 96.60 ± 5.74 | 95.67 ± 8.00 | 95.63 ± 8.04 | 72.67 |
| EfficientNet-B0 + Log-mel | 95.20 ± 7.92 | 96.30 ± 5.94 | 95.20 ± 7.92 | 95.15 ± 8.15 | 74.67 |

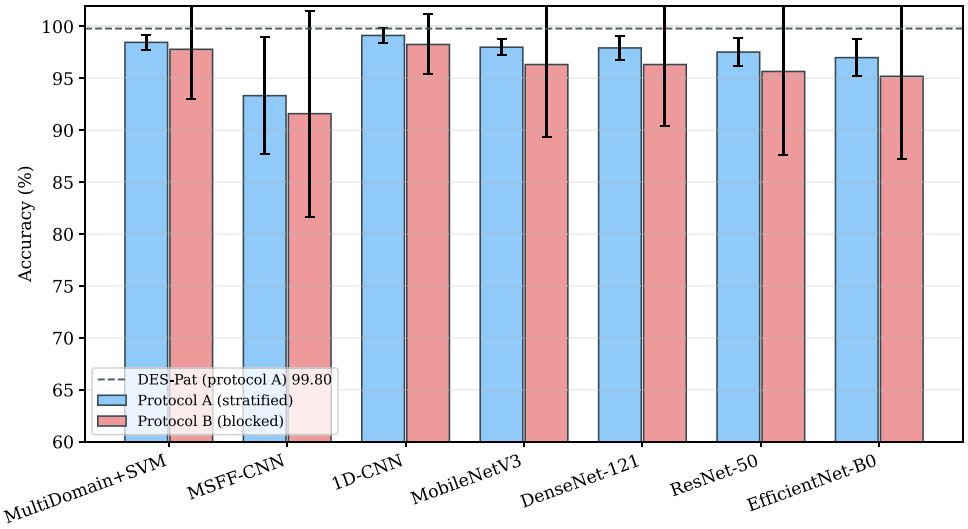



**Figure 5:** Accuracy of all re-executed methods under both protocols.

Per-class performance is reported in Table 6. Fig. 6 shows the test confusion matrices of the median-accuracy fold under each protocol. The median fold is shown rather than the best fold so that the illustration is not selected optimistically. Protocol A produces isolated errors, whereas Protocol B concentrates them in adjacent blade counts.

**Table 6:** Per-class performance (%) of the proposed EfficientNet-B0 + log-mel, averaged over 10 folds, under both protocols.

| Propeller Class | Precision | Recall | F1-Score |
|---|---|---|---|
| *Protocol A: stratified* | | | |
| Two blades | 97.46 ± 4.47 | 95.33 ± 3.40 | 96.32 ± 3.21 |
| Three blades | 98.63 ± 2.24 | 97.67 ± 3.67 | 98.14 ± 2.88 |
| Four blades | 97.11 ± 2.65 | 98.00 ± 2.21 | 97.52 ± 1.69 |
| Five blades | 99.67 ± 1.00 | 98.67 ± 1.63 | 99.16 ± 1.12 |
| Six blades | 92.58 ± 3.22 | 95.33 ± 4.76 | 93.89 ± 3.50 |
| *Protocol B: blocked* | | | |
| Two blades | 95.95 ± 9.86 | 95.00 ± 5.82 | 95.27 ± 7.35 |
| Three blades | 99.38 ± 1.88 | 93.33 ± 18.92 | 94.87 ± 13.78 |
| Four blades | 98.08 ± 4.79 | 93.33 ± 11.06 | 95.42 ± 7.77 |
| Five blades | 98.35 ± 3.06 | 96.33 ± 8.23 | 97.13 ± 5.10 |
| Six blades | 89.74 ± 15.11 | 98.00 ± 3.06 | 93.06 ± 10.01 |

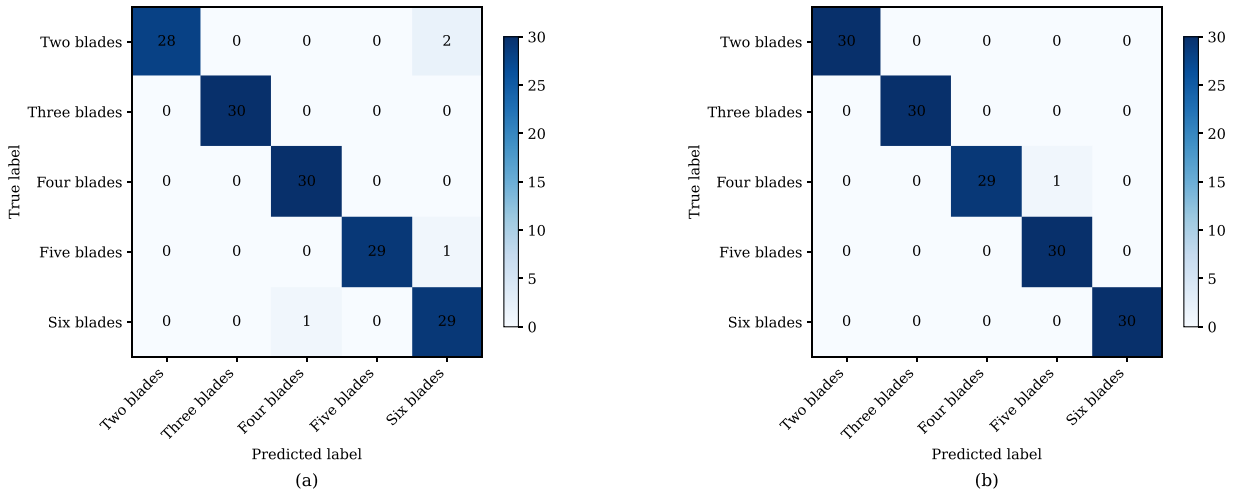



**Figure 6:** Test confusion matrices of the proposed model on the median-accuracy fold of each protocol. (**a**) Protocol A (stratified), fold 4, accuracy 97.33%. (**b**) Protocol B (blocked), fold 6, accuracy 99.33%.

The five-blade class is the most reliable under both protocols. The six-blade class shows the lowest precision, with confusions toward its adjacent blade counts. Under the blocked protocol the per-class standard deviations grow substantially. The three-blade recall drops to 93.33 ± 18.92%. The wide deviations reflect the weak first and last temporal blocks rather than a uniform error rate.

### 5.2 Ablation Studies

The ablation study isolates the factors that plausibly drive the reported accuracy. Each variant is a complete Protocol A 10-fold sweep of the EfficientNet-B0 pipeline with one factor changed. The random-initialisation variant is the one exception. Freezing an untrained backbone would fit the head to random projections, so that variant uses the single-phase schedule of the from-scratch models. Paired tests against the unmodified baseline use the shared folds. That baseline is the proposed pipeline, with log-mel input over 128 bands, ImageNet initialisation, the two-layer head, and all four augmentation operations. The reported Δ is the accuracy change against it in percentage points. The $p$-values come from paired t-tests and Wilcoxon signed-rank tests over the shared ten folds. The segment-length ablation was not re-executed. The 1 s segmentation is fixed by the source protocol of Yaman et al. [18]. Re-segmenting would also change the sample counts of every other experiment. Table 7 reports the results.

**Table 7:** Ablation results for the proposed pipeline under Protocol A.

| Variant | ACC (%) | UAP (%) | UAR (%) | F1 (%) | Min | Δ | $p_t$ | $p_W$ |
|---|---|---|---|---|---|---|---|---|
| Baseline (proposed pipeline) | 97.00 ± 1.77 | 97.09 ± 1.74 | 97.00 ± 1.77 | 97.01 ± 1.77 | 92.67 | – | – | – |
| *Input representation* | | | | | | | | |
| Linear mel, no log compression | 77.20 ± 13.35 | 79.07 ± 15.00 | 77.20 ± 13.35 | 75.05 ± 15.66 | 54.00 | −19.80 | 0.001 | 0.002 |
| 64 mel bands | 95.47 ± 1.42 | 95.65 ± 1.37 | 95.47 ± 1.42 | 95.47 ± 1.40 | 93.33 | −1.53 | 0.014 | 0.012 |
| *Pretraining* | | | | | | | | |
| Random init, no pretraining | 97.53 ± 1.43 | 97.66 ± 1.33 | 97.53 ± 1.43 | 97.54 ± 1.42 | 94.00 | +0.53 | 0.168 | 0.281 |
| *Classification head* | | | | | | | | |
| Single linear layer | 97.07 ± 1.24 | 97.18 ± 1.16 | 97.07 ± 1.24 | 97.07 ± 1.23 | 94.67 | +0.07 | 0.910 | 0.883 |
| Hidden layer, no dropout | 97.67 ± 1.00 | 97.76 ± 0.96 | 97.67 ± 1.00 | 97.67 ± 0.99 | 96.00 | +0.67 | 0.213 | 0.344 |
| *Augmentation* | | | | | | | | |
| No augmentation | 97.47 ± 1.33 | 97.58 ± 1.26 | 97.47 ± 1.33 | 97.48 ± 1.31 | 94.67 | +0.47 | 0.242 | 0.422 |
| Time stretch only | 96.87 ± 1.27 | 96.93 ± 1.25 | 96.87 ± 1.27 | 96.87 ± 1.26 | 94.67 | −0.13 | 0.805 | 0.688 |
| Additive noise only | 96.73 ± 1.44 | 96.86 ± 1.40 | 96.73 ± 1.44 | 96.75 ± 1.43 | 94.67 | −0.27 | 0.657 | 0.562 |
| Circular shift only | 96.73 ± 2.30 | 96.88 ± 2.18 | 96.73 ± 2.30 | 96.75 ± 2.28 | 91.33 | −0.27 | 0.653 | 0.438 |
| SpecAugment only | 97.40 ± 1.80 | 97.52 ± 1.71 | 97.40 ± 1.80 | 97.41 ± 1.80 | 92.67 | +0.40 | 0.260 | 0.375 |

Two choices in the input representation are significant. First, log compression is the single most important design choice in the pipeline. Removing it costs 19.8 percentage points on average and destabilises training. Individual folds fall to 54%. Log compression amplifies the low-energy blade-pass harmonics that the saliency analysis of Section 5.4 identifies as the decision-relevant structure. Second, 128 mel bands outperform 64 bands by a significant 1.53 percentage points, confirming that the harmonic ladder needs adequate frequency resolution.

No other factor tested here reaches significance. ImageNet pretraining is not required on this dataset. A randomly initialised backbone reaches 97.53 ± 1.43%, which sits 0.53 points above the pretrained baseline and is not significantly different ($p = 0.168$). The design of the classification head is equally immaterial. A single linear layer gives 97.07 ± 1.24% and removing dropout gives 97.67 ± 1.00%, at $p = 0.910$ and $p = 0.213$. Augmentation does not improve accuracy in any of the configurations tested. The four operations define 16 possible subsets of enabled operations, and six of them were evaluated. These are the empty set, each operation alone, and the full set used by the baseline. Measured against the unaugmented run, each of the

three waveform operations costs between 0.60 and 0.73 points, while SpecAugment is neutral. None of those differences is significant at the 5% level. Their direction is nevertheless consistent, which suggests that the waveform operations add variation the task does not contain. The six evaluated subsets lie between 96.73% and 97.47%. This spread of 0.73 points is less than half the fold standard deviation of the baseline. The ten subsets with two or three operations were not evaluated, so interactions between operations remain untested.

### *5.3 Training Dynamics and Computational Efficiency*

Fig. 7 shows the measured training curves, averaged over the ten Protocol A folds. The transfer models converge within roughly ten fine-tuning epochs. The from-scratch MSFF-CNN converges more slowly and less reliably, which matches its weaker fold statistics.

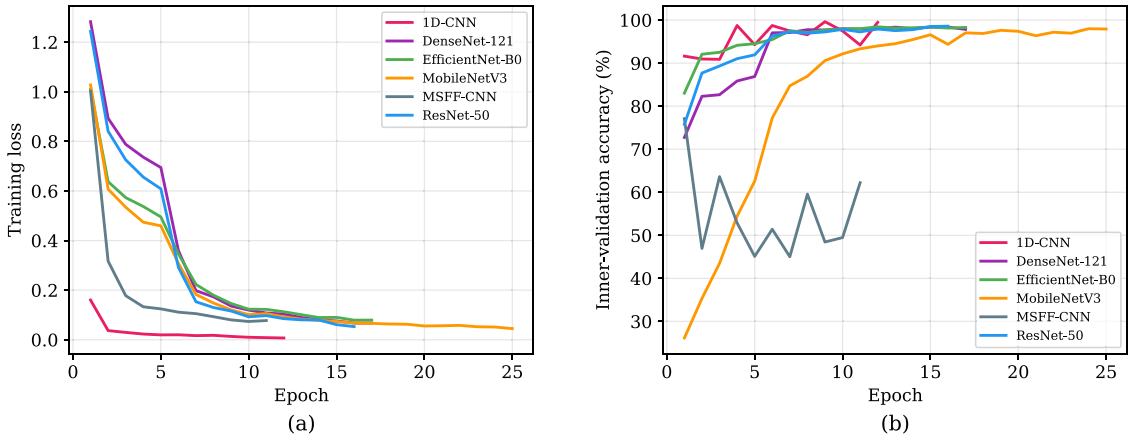



**Figure 7:** Measured training dynamics averaged over the ten Protocol A folds. (**a**) Training loss. (**b**) Inner-validation accuracy.

Table 8 lists the measured computational requirements. Training times are wall-clock means per fold on the hardware of Section 4, including data loading and augmentation. Latency is the mean single-sample inference time after warm-up, measured on the device used for training. The spectrogram models ran on the GPU and the 1D-CNN on the CPU, so latencies are indicative rather than directly comparable. MSFF-CNN has the lowest latency, and MobileNetV3 is the fastest transfer model. These results, taken with the accuracy outcomes, inform the deployment discussion in Section 6.

**Table 8:** Measured computational comparison of all deep learning models under Protocol A.

| Model | Params (M) | Device | Train (s/Fold) | Latency (ms) |
|---|---|---|---|---|
| ResNet-50 | 24.03 | GPU | 1030 ± 223 | 17.5 ± 1.7 |
| EfficientNet-B0 | 4.34 | GPU | 786 ± 199 | 17.1 ± 1.1 |
| MobileNetV3 | 3.22 | GPU | 911 ± 85 | 8.3 ± 1.7 |
| DenseNet-121 | 7.22 | GPU | 1016 ± 245 | 21.0 ± 1.3 |
| MSFF-CNN | 0.37 | GPU | 568 ± 252 | 2.4 ± 0.3 |
| 1D-CNN | 0.46 | CPU | 1085 ± 342 | 9.9 ± 6.1 |

### *5.4 Physical Interpretation of the Learned Features*

A feature-attribution analysis addresses the black-box behaviour of the trained blocked-protocol EfficientNet-B0 model. Fig. 8 shows Grad-CAM saliency maps for a representative segment of each class. Fig. 9a reduces the saliency to a mean profile per mel band, averaged over ten segments per class.

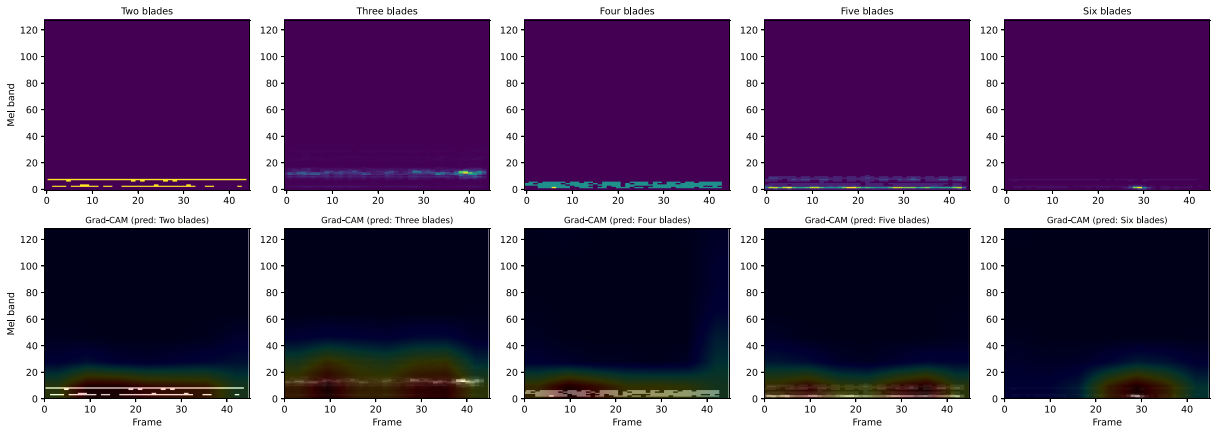



**Figure 8:** Grad-CAM analysis of the blocked-protocol model, with log-mel-spectrograms above and saliency overlays below.

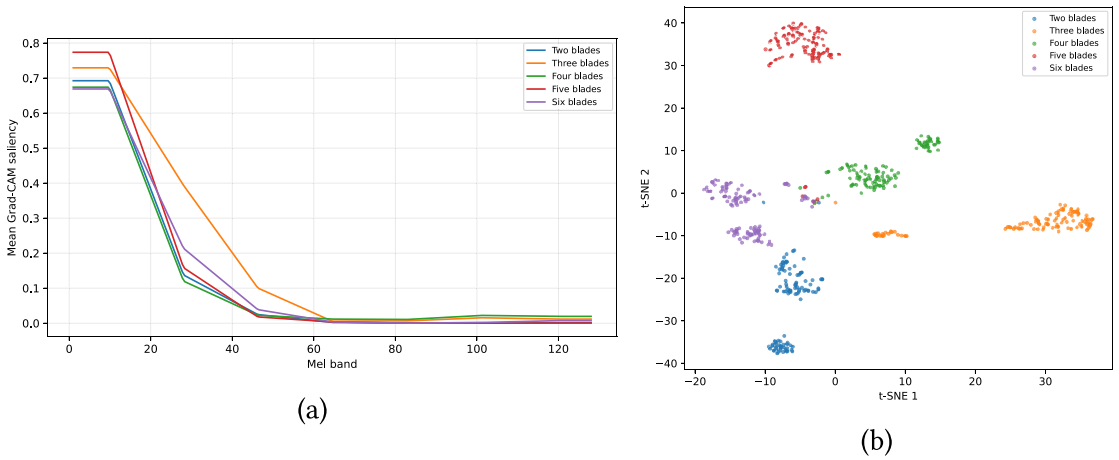



**Figure 9:** Feature analysis of the proposed model. (**a**) Mean Grad-CAM saliency per mel band. (**b**) t-SNE embedding of penultimate features.

Three observations follow. First, the saliency concentrates almost entirely below mel band 40, which corresponds to frequencies below approximately 1 kHz. This is the region occupied by the blade-pass frequency and its harmonic ladder in Fig. 1. The broadband high-frequency region associated with cavitation noise contributes almost nothing to the decisions. The network therefore reads the tonal harmonic structure rather than the broadband floor. Second, the saliency profiles are class-dependent. The three-blade class holds attention up to approximately mel band 50, consistent with the wider harmonic spacing visible in its spectrogram. The remaining classes concentrate below band 30. Third, the median segment of every class is correctly classified by the blocked-protocol model, and the attended regions coincide with the visible harmonic lines.

Fig. 9b shows a t-distributed stochastic neighbour embedding (t-SNE) projection of the penultimate features of every third segment. The five classes form well-separated clusters. Each class additionally fragments into two or three sub-clusters. This fragmentation indicates distinct temporal regimes within each single recording, which is consistent with the weak first and last folds of the blocked protocol. A small mixed region near the origin contains segments from several classes; these are the hard segments that dominate the blocked-protocol errors.

These findings support the claim that the network learns blade-count-related harmonic structure rather than arbitrary texture.

## 6 Discussion

### *6.1 Interpretation of the Benchmark Results*

The re-executed benchmark supports three interpretations. First, the evaluation protocol shapes the conclusions more strongly than the architecture. Under segment-level cross-validation with test-fold checkpoint selection, near-perfect accuracies are attainable on this dataset. Under inner-validation model selection the same models score 93%–99%, and under the blocked protocol their stability collapses on the recording edges. Any ranking claimed on this dataset therefore depends on the protocol, and the protocol must be reported with the same care as the model.

Second, the raw-waveform 1D-CNN outperforms every spectrogram model under both protocols. Two mechanisms plausibly contribute. The spectrogram pipeline compresses each segment to a fixed image and discards phase, while the waveform network can exploit fine temporal structure directly. In addition, the small parameter count acts as a strong regulariser at this data scale. The model also depends on no ImageNet prior that may fit tank-specific texture. The Grad-CAM analysis of Section 5.4 shows that the spectrogram models do read the blade-pass harmonic region. Reading the right region is evidently not sufficient for temporal robustness.

Third, the comparison with DES-Pat [18] must be protocol-matched. The published 99.80% was obtained under a segment-level protocol on the same recordings. None of the re-executed models reaches that level under inner-validation model selection, and the blocked protocol lowers the attainable level further. The structural advantages of learned representations over cipher-derived descriptors remain, namely end-to-end training and adaptability to new tasks. The claimed accuracy advantage, however, does not survive leakage-aware evaluation, and this study reports that outcome openly.

The MSFF-CNN (93.33%, 0.37M parameters) sets the from-scratch lower bound. Its weakness reflects the capacity of its architecture rather than the missing ImageNet prior, as the random-initialisation ablation of Section 5.2 shows. The multi-domain SVM baseline (98.47%) confirms that carefully engineered classical features remain highly competitive on controlled laboratory data.

### *6.2 Practical Deployment*

The benchmark results point to a single primary recommendation. The 1D-CNN combines the highest accuracy under both protocols, the best worst-fold behaviour, a 0.46M parameter budget, and no spectrogram stage. It is therefore the preferred candidate for both accuracy-driven and resource-constrained applications on this task. MobileNetV3 is the strongest spectrogram alternative when a pretrained backbone is preferred. Three qualifications now bound these recommendations. First, all latency figures in Table 8 are PyTorch measurements on the desktop hardware of Section 4, not on embedded devices. Second, the figures cover model inference only. They exclude microphone acquisition, buffering, spectrogram computation, memory footprint, and energy consumption, all of which matter on an autonomous underwater vehicle.

Third, the blocked-protocol results show that accuracy under temporal distribution shift is markedly less stable than the segment-level numbers suggest. Deployment on embedded sonar hardware therefore remains a hypothesis to be validated, and it is listed as future work rather than claimed as a capability.

### *6.3 Limitations*

Several limitations constrain the scope of the present results. All recordings were made in a controlled laboratory tank. This setup suppresses multipath propagation, depth-dependent sound-speed gradients, and shipping noise. Such effects characterise open-sea deployments. Whether the learned spectro-temporal features remain discriminative under these conditions is an open question. The dataset was collected at a single motor voltage. This keeps shaft speed roughly constant during recording. Operational propellers span a wide speed range. Speed variation shifts blade-pass frequencies and alters harmonic spacing in ways not represented in the training data. The 1,500-sample corpus is also modest by deep learning standards. The uniformly high accuracy across architectures suggests the task may be simpler than that imposed by real-world deployments. This limits the strength of generalisability claims. A systematic analysis of augmentation combinations was not performed. Only six of the 16 subsets of the four operations were evaluated, so interactions between operations remain untested. Finally, the 3D-printed test propellers (30 mm diameter) are geometric analogues rather than scaled replicas of marine hardware. Full-scale propellers produce different cavitation regimes and structural resonances. Direct transfer to operational vessels requires dedicated experimental validation at representative scales.

## 7 Conclusion

This study re-evaluated deep learning for underwater propeller blade-count recognition under a leakage-aware experimental design. All experiments were re-executed with inner-validation model selection, and every result is backed by machine-readable per-fold artifacts. Under the established segment-level protocol, no model reproduces the near-perfect accuracies previously reported on this dataset. The raw-waveform 1D-CNN is the best model at $99.13 \pm 0.73\%$, and its advantage over the transfer models is statistically significant. Under the blocked recording-level protocol, the same 1D-CNN remains the most robust at $98.27 \pm 2.89\%$. The four ImageNet transfer models fall to 72%–79% on the first temporal block, which segment-level evaluation conceals entirely. The interpretability analysis shows that the spectrogram networks attend to the blade-pass harmonic region below 1 kHz. The ablations establish log compression as the dominant design factor, worth 19.8 percentage points, and 128 mel bands as significantly better than 64. Taken together, the results support one central conclusion. On single-recording-per-class datasets, the evaluation protocol governs the reported accuracy more than the classifier, and protocol details must be reported alongside every result.

The future work will aim to validate the framework on open-sea recordings. Such recordings introduce realistic multipath propagation and ambient noise. Additional directions include multi-speed propeller recognition with temporal attention mechanisms. Self-supervised pretraining on unlabelled underwater acoustic data is another promising direction. A full factorial evaluation of the four augmentation operations would resolve the interactions that the present ablation leaves untested. Embedded hardware deployment on autonomous underwater vehicle platforms remains a priority for real-time applications.

**Acknowledgement:** The author thanks Yaman et al. for making the propeller acoustic dataset publicly available.

**Funding Statement:** The author received no specific funding for this study.

**Availability of Data and Materials:** The propeller acoustic dataset analysed in this study is openly available in the Propeller repository at https://github.com/orhanyaman/Propeller. The repository is maintained by the dataset authors and provides five audio recordings, one per blade-count class. The decoded recordings last 323, 313, 375, 348, and 340 s for the two- to six-blade classes at a 44 kHz source rate. The analysis code, the per-fold results in machine-readable form, and the trained model checkpoints are permanently archived on Zenodo at https://doi.org/10.5281/zenodo.22803169 (accessed on 17 September 2026).

**Ethics Approval:** Not applicable. This study did not involve human participants or animals.

**Conflicts of Interest:** The author declares no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| 1D | One-Dimensional |
| AAC | Advanced Audio Coding |
| ACC | Accuracy |
| AGWT | Adaptive Generalized Warblet Transform |
| BPF | Blade-Pass Frequency |
| CNN | Convolutional Neural Network |
| CPU | Central Processing Unit |
| CUDA | Compute Unified Device Architecture |
| DC | Direct Current |
| DCMLBP | Dynamic Center Mirror Local Binary Pattern |
| DEMON | Detection of Envelope Modulation on Noise |
| DES | Data Encryption Standard |
| FFT | Fast Fourier Transform |
| GAP | Global Average Pooling |
| GPU | Graphics Processing Unit |
| Grad-CAM | Gradient-Weighted Class Activation Mapping |
| HHT | Hilbert–Huang Transform |
| HHT-INR | HHT with Intelligent Noise Reduction |
| KNN | K-Nearest Neighbours |
| MBConv | Mobile Inverted Bottleneck Convolution |
| MFCC | Mel-Frequency Cepstral Coefficients |
| MLP | Multi-Layer Perceptron |
| MSFF | Multi-Scale Feature Fusion |
| NCA | Neighbourhood Component Analysis |
| PCM | Pulse-Code Modulation |
| RAM | Random-Access Memory |
| ReLU | Rectified Linear Unit |
| RMS | Root-Mean-Square |
| S-Box | Substitution Box |
| SE | Squeeze-and-Excitation |
| SNR | Signal-to-Noise Ratio |
| STFT | Short-Time Fourier Transform |
| SVM | Support Vector Machine |
| t-SNE | t-Distributed Stochastic Neighbour Embedding |
| UAP | Unweighted Average Precision |
| UAR | Unweighted Average Recall |
| WGAN | Wasserstein Generative Adversarial Network |

## References


1. Lampe T, Radtke L, Abdel-Maksoud M, Düster A. A partitioned solution approach for the simulation of dynamic behaviour and acoustic signature of flexible cavitating marine propellers. Ocean Eng. 2020;197:106854. doi:10.1016/j.oceaneng.2019.106854.
2. Carlton JS. Marine propellers and propulsion. 4th ed. Oxford, UK: Butterworth-Heinemann; 2019.
3. Becker G, Güdesen A. Passive sensing with acoustics on the battlefield. Appl Acoust. 2000;59(2):149–78. doi:10.1016/s0003-682x(99)00023-7.
4. Prabowo AR, Tuswan T, Ridwan R. Advanced development of sensors' roles in maritime-based industry and research: from field monitoring to high-risk phenomenon measurement. Appl Sci. 2021;11(9):3954. doi:10.3390/app11093954.
5. Song Y, Liu J, Cao L, Chu N, Wu D. Robust passive underwater acoustic detection method for propeller. Appl Acoust. 2019;148:151–61. doi:10.1016/j.apacoust.2018.12.024.
6. Korkut E, Atlar M. An experimental investigation of the effect of foul release coating application on performance, noise and cavitation characteristics of marine propellers. Ocean Eng. 2012;41:1–12. doi:10.1016/j.oceaneng.2011.12.012.
7. Kimmerl J, Mertes P, Abdel-Maksoud M. Application of large eddy simulation to predict underwater noise of marine propulsors. Part 1: cavitation dynamics. J Mar Sci Eng. 2021;9(8):792. doi:10.3390/jmse9080792.
8. Kimmerl J, Mertes P, Abdel-Maksoud M. Application of large eddy simulation to predict underwater noise of marine propulsors. Part 2: noise generation. J Mar Sci Eng. 2021;9(7):778. doi:10.3390/jmse9070778.
9. Zhu C, Gaggero T, Makris NC, Ratilal P. Underwater sound characteristics of a ship with controllable pitch propeller. J Mar Sci Eng. 2022;10(3):328. doi:10.3390/jmse10030328.
10. Recuero Virto L, Dumez H, Romero C, Bailly D. How can ports act to reduce underwater noise from shipping? Identifying effective management frameworks. Mar Pollut Bull. 2022;174(9):113136. doi:10.1016/j.marpolbul.2021.113136.
11. Aslam MA, Zhang L, Liu X, Irfan M, Xu Y, Li N, et al. Underwater sound classification using learning based methods: a review. Expert Syst Appl. 2024;255:124498.
12. Filho EPS, Santos AD, Simas Filho EF, Fernandes ACL Jr, de Seixas JM, de Moura NN Jr. Hilbert–Huang transform with intelligent noise reduction for passive SONAR signal processing. IEEE J Ocean Eng. 2025;50(2):1387–402. doi:10.1109/joe.2024.3519737.
13. Wang M, Zheng C, Sun D, Na T. Measurement and application of underwater acoustic low-frequency line spectral signal micro-Doppler features based on Adaptive Generalized Warblet Transform. Measurement. 2026;261(4):119985. doi:10.1016/j.measurement.2025.119985.
14. Khishe M, Mosavi MR. Classification of underwater acoustical dataset using neural network trained by Chimp Optimization Algorithm. Appl Acoust. 2020;157(13):107005. doi:10.1016/j.apacoust.2019.107005.
15. González-Hernández FR, Sánchez-Fernández LP, Suárez-Guerra S, Sánchez-Pérez LA. Marine mammal sound classification based on a parallel recognition model and octave analysis. Appl Acoust. 2017;119(2):17–28. doi:10.1016/j.apacoust.2016.11.016.
16. Khishe M, Safari A. Classification of sonar targets using an MLP neural network trained by dragonfly algorithm. Wirel Pers Commun. 2019;108(4):2241–60. doi:10.1007/s11277-019-06520-w.
17. Tuncer T, Aydemir E, Dogan S. Automated ambient recognition method based on dynamic center mirror local binary pattern: DCMLBP. Appl Acoust. 2020;161:107165. doi:10.1016/j.apacoust.2019.107165.
18. Yaman O, Tuncer T, Tasar B. DES-Pat: a novel DES pattern-based propeller recognition method using underwater acoustical sounds. Appl Acoust. 2021;175:107859. doi:10.1016/j.apacoust.2020.107859.
19. Yang S, Xue L, Hong X, Zeng X. A lightweight network model based on an attention mechanism for ship-radiated noise classification. J Mar Sci Eng. 2023;11(2):432. doi:10.3390/jmse11020432.
20. Liu F, Li Z, Yang K, Chen F, Yu J. MmPiFNN: a multi-mode physics-informed fuzzy neural network for passive recognition of surface ships by underwater equipment using ship radiated noise signals. Def Technol. 2026;59(4):243–66. doi:10.1016/j.dt.2025.11.003.

21. Karabacak YE. Deep learning-based CNC milling tool wear stage estimation with multi-signal analysis. Maint Reliab. 2023;25(3):168082. doi:10.17531/ein/168082.
22. Hershey S, Chaudhuri S, Ellis DPW, Gemmeke JF, Jansen A, Moore RC, et al. CNN architectures for large-scale audio classification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017 Mar 5–9. New Orleans, LA, USA. p. 131–5.
23. Zhao Z, Alzubaidi L, Zhang J, Duan Y, Gu Y. A comparison review of transfer learning and self-supervised learning: definitions, applications, advantages and limitations. Expert Syst Appl. 2024;242(10):122807. doi:10.1016/j.eswa.2023.122807.
24. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 770–8.
25. Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 2261–9.
26. Howard A, Sandler M, Chen B, Wang W, Chen LC, Tan M, et al. Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 1314–24.
27. Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML); 2019 Jun 9–15; Long Beach, CA, USA. p. 6105–14.
28. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR); 2015 May 7–9; San Diego, CA, USA.
29. Park DS, Chan W, Zhang Y, Chiu CC, Zoph B, Cubuk ED, et al. SpecAugment: a simple data augmentation method for automatic speech recognition. In: Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech); 2019 Sep 15–19; Graz, Austria. p. 2613–7.
30. Gong Y, Chung YA, Glass J. AST: audio spectrogram transformer. In: Proceedings of the 22nd Annual Conference of the International Speech Communication Association (Interspeech); 2021 Aug 30–Sep 3; Brno, Czechia. p. 571–5.
31. Tian SZ, Chen DB, Fu Y, Zhou JL. Joint learning model for underwater acoustic target recognition. Knowl Based Syst. 2023;260(4):110119. doi:10.1016/j.knosys.2022.110119.
32. Wang Z, Li Z, Yang K. PC-LNN: a physics-constrained liquid neural network for multi-scale ship radiated noise recognition. Meas Sci Technol. 2026;37(29):296101. doi:10.1088/1361-6501/ae887b.