Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 1, January 2026, pp: 168-184 e-ISSN: 2656-8632

RESEARCH ARTICLE

OPEN ACCESS

# Deep Learning-Based Lung Sound Classification Using Mel-Spectrogram Features for Early Detection of Respiratory Diseases

**Midfai Yabani[1], Mohammad Reza Faisal[1], Fatma Indriani[1], Dodon Turianto Nugrahadi[1], Dwi Kartini[1], and Kenji Satou[2]**

[1] Department of Computer Science, Lambung Mangkurat University, Banjarbaru, Indonesia
[2] Faculty of Transdisciplinary Sciences for Innovation, Kanazawa University, Kanazawa, Japan

**Corresponding author**: Mohammad Reza Faisal. (e-mail: reza.faisal@ulm.ac.id), **Author(s) Email**: Midfai Yabani (e-mail: midfaiybni@gmail.com), Fatma Indriani (e-mail: f.indriani@ulm.ac.id), Dodon Turianto Nugrahadi (e-mail: dodonturianto@ulm.ac.id), Dwi Kartini (e-mail: dwikartini@ulm.ac.id), Kenji Satou (e-mail: ken@t.kanazawa-u.ac.jp)

**Abstract** **Respiratory diseases such as asthma, chronic obstructive pulmonary disease, and pneumonia remain among the leading causes of death globally. Traditional diagnostic approaches, including auscultation, rely heavily on the subjective expertise of medical practitioners and the quality of the instruments used. Recent advancements in artificial intelligence offer promising alternatives for automated lung sound analysis. However, audio is an unstructured data format that must be converted into a suitable format for AI algorithms. Another significant challenge lies in the imbalanced class distribution within available datasets, which can adversely affect classification performance and model reliability. This study applied several comprehensive preprocessing techniques, including random undersampling to address data imbalance, resampling audio at 4000 Hz for standardization, and standardizing audio duration to 2.7 seconds for consistency. Feature extraction was then performed using the Mel Spectrogram method, converting audio signals into image representations to serve as input for classification algorithms based on deep learning architectures. To determine optimal performance characteristics, various Convolutional Neural Network (CNN) architectures were systematically evaluated, including LeNet-5, AlexNet, VGG-16, VGG-19, ResNet-50, and ResNet-152. VGG-16 achieved the highest classification accuracy of the tested models at 75.5%, demonstrating superior performance in respiratory sound classification tasks. This study demonstrates the potential of AI-based lung sound classification systems as a complementary diagnostic tool for healthcare professionals and the general public in supporting early identification of respiratory abnormalities and diseases. The findings suggest that automated lung sound analysis could enhance diagnostic accessibility and provide more valuable support for clinical decision-making in respiratory healthcare applications.**

**Keywords** **Lung Sound; Feature Extraction; Au; Audio Classification; Convolutional Neural Network**.

## I. Introduction

Lung sound classification is crucial in medical diagnostics, particularly in identifying respiratory conditions such as asthma, chronic obstructive pulmonary disease (COPD), and pneumonia [1], [2]. These conditions are often characterized by abnormal lung sounds, such as crackles and wheezes, which can be detected through auscultation [3], [4]. However, manual interpretation of lung sounds requires specialized expertise and is often prone to inter-observer variability, which may lead to inconsistent diagnostic outcomes [5], [6]. These challenges have motivated the development of automated lung sound classification systems to support healthcare professionals and the general public in the early detection of respiratory diseases. Recent advancements in artificial intelligence (AI) have provided promising alternative solutions for audio analysis, including lung sound classification [7], [8]. Classifying lung sounds using AI generally follows the same pipeline as in other audio-based tasks, beginning with extracting raw audio into a structured format that machine learning algorithms can process. Commonly used feature extraction methods include spectral features, such as Mel-Frequency Cepstral Coefficients (MFCC), and time–frequency representations, such as spectrograms [9], [10], [11]. MFCC is favored for its computational efficiency and robustness to moderate noise levels [11], [12]. However, due to dimensionality reduction, MFCC may discard critical information,

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

leading to lower classification performance than more expressive representations like Mel-Spectrograms. Mel-spectrograms offer a richer representation by capturing key characteristics of audio signals, including relevant frequency patterns and intensity variations, making them particularly useful for medical classification tasks [13], [14], [15]. MFCC-based feature extraction typically yields numerical feature vectors that can be fed into traditional machine learning algorithms such as Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbors (KNN), and Random Forest [16]. In contrast, feature extraction methods based on Gammatonegrams or spectrograms produce image-like representations that are not directly compatible with traditional algorithms but are well-suited for processing with deep learning–based classifiers, particularly Convolutional Neural Networks (CNNs) [7], [9], [13]. To address this, numerous CNN architectures have been developed and optimized for image-based inputs, including VGG-16, ResNet-50, MobileNet, and GoogLeNet.

Respiratory sound analysis research for pulmonary disease diagnosis commonly utilizes the ICBHI 2017 dataset (International Conference on Biomedical and Health Informatics), which consists of four classes. Studies employing Gammatonegram-based feature extraction on the ICBHI 2017 dataset, combined with two Convolutional Neural Network (CNN) architectures, ResNet-50 and VGG-16, reported classification accuracies of 60.80% and 67.97%, respectively [17], [18]. Using the same classification algorithms with Mel Spectrogram feature extraction methods, reported accuracies ranged between 60.80%–62.29% and 62.50%–67.97% [17], [18]. Other studies using GoogLeNet and MobileNet architectures with Mel Spectrogram input achieved accuracies of 63.69% [18] and 74% [19], respectively. In addition to the CNN architectures mentioned above, models such as LeNet-5, AlexNet, VGG-19, and ResNet-152 have been successfully applied to various audio analysis tasks. For instance, VGG-19 combined with Mel Spectrogram has demonstrated effective performance in the classification of COVID-19 based on cough sounds [20]. Similarly, LeNet-5 has shown strong performance in medical and mechanical audio classification tasks, such as breast cancer diagnosis and faulty motor sound detection [21], [22]. Moreover, AlexNet has been effectively utilized in cardiac anomaly detection, outperforming several other classification methods [23].

Prior studies on lung sound classification using the ICBHI 2017 dataset have demonstrated classification accuracies ranging from 60% to 74%, indicating a clear opportunity for further exploration to improve diagnostic performance. Building upon this gap, the primary objective of this study is to systematically evaluate and compare the performance of six different CNN architectures, LeNet-5, AlexNet, VGG-16, VGG-19, ResNet-50, and ResNet-152, in combination with Mel Spectrogram-based feature extraction, to identify the most effective model for automated lung sound classification. The novelty of this research lies in its integrated methodological enhancements, including:

1. Class balancing through random undersampling to address data imbalance.
2. Audio preprocessing involving resampling at 4000 Hz and segment duration standardization (2.7 seconds), and
3. Hyperparameter variation, such as epochs, learning rate, optimizer choice, batch size, and input shape tuning, is used to optimize model training.

Unlike previous studies that focus on a single architecture or a limited set of configurations, this research comprehensively compares architectures under controlled preprocessing conditions. The findings are expected to contribute to higher diagnostic accuracy and greater reproducibility, and to improve objectivity in computer-aided respiratory disease detection systems, ultimately supporting both clinical decision-making and public health applications.

## II. Method

### A. Dataset

In this study, the respiratory sound data were obtained from the ICBHI 2017 dataset, one of the largest publicly available collections of respiratory audio recordings. The dataset contains a total of 5.5 hours of audio, comprising 6,898 respiratory cycles, including 3,642 normal cycles, 1,864 cycles with crackles, 886 cycles with wheezes, and 506 cycles containing a combination of crackles and wheezes [24]. Each audio file in the dataset is accompanied by metadata that provides detailed information about the recording conditions, including the equipment used, the acquisition mode, and the chest location of the recording. The equipment used to acquire lung sound recordings include an AKG C417L Microphone, a 3M Littmann Classic II SE Stethoscope, a 3M Littmann 3200 Electronic Stethoscope, and a Welch Allyn Meditron Master Elite Electronic Stethoscope. The recordings were obtained using two acquisition modes: sequential (single-channel) and simultaneous (multi-channel). The chest locations at which the sounds were recorded include the trachea and the anterior left, anterior right, posterior left, posterior right, lateral left, and lateral right positions. Additional patient information, such as age, sex, and whether the subject is an adult or child, is also provided in the dataset.

### B. Method

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

This study employs a quantitative approach to data analysis and applies convolutional neural networks (CNNs) to classify respiratory sounds. Fig. 1. presents the overall workflow of the proposed methodology. The first step is data collection. The second step involves audio resampling. The audio samples in the ICBHI 2017 dataset were recorded at various sampling rates, ranging from 4,000 Hz to 44,100 Hz. This study focuses on four classes: crackle, wheeze, normal, and both (crackle and wheeze). Crackle sounds typically occur within the 100–700 Hz frequency range [25], [26], while wheeze sounds fall within 100–1000 Hz [27], [26]. Since the relevant signal frequencies are generally below 2000 Hz, resampling the audio to 4,000 Hz is sufficient to preserve important signal characteristics [26]. For example, if the original sampling rate $f_s^{(old)}$ is 44,100 Hz and the target sample rate $f_s^{(new)}$ is 4,000 Hz, the resampling process can be mathematically defined in Eq. (1) [28] as:

$$y[m] = x(\frac{f_s^{(old)}}{f_s^{(new)}} m) \quad (1)$$

Substituting the values gives y[m] = x(11.025 · m) where the ratio $f_s^{(old)}$ divided by $f_s^{(new)}$ indicates that each new sample in the resampled signal y[m] corresponds to a position 11.025 samples apart in the original signal x[n]. Since this ratio is non-integer, interpolation is applied to estimate values of x at fractional indices, resulting in a new signal with a sampling rate of 4000 Hz while preserving the original signal's duration and content. The third preprocessing step is segmentation, in which each resampled audio file is segmented into individual respiratory cycles labeled as crackle, wheeze, normal, or both, based on expert annotations. However, the duration of respiratory cycles varies significantly, ranging from 0.2 to 16 seconds, with an average of approximately 2.7 seconds. A twofold strategy is employed to standardize input duration: cycles longer than 2.7 seconds are truncated to retain only the first 2.7 seconds. In contrast, shorter cycles are padded using sample-level zero-padding [28]. The final preprocessing step is encoding, where class names are converted into corresponding numerical labels.

The next phase is feature extraction, aimed at transforming audio data into structured input suitable for classification algorithms. This study utilizes the Mel-Spectrogram, which converts audio signals into the time–frequency domain by mapping frequency to the Mel scale, a perceptual scale of pitches more aligned with human auditory perception [13], [29], [30]. To obtain the frequency domain representation, each windowed segment of the audio signal x[n] is transformed using the Fast Fourier Transform (FFT), which is defined in Eq. (2) [32].

$$X[k] = \sum_{n=0}^{n-1} x[n]w[n]e^{-j2\pi kn/N} \quad (2)$$

where *w[n]* denotes the window function, N is the window length, and *X[k]* represents the complex spectrum for frequency bin *k*. The power spectrum is computed as Eq. (3) [32].

$$P[k] = |X[k]|^2 \quad (3)$$

Next, the Mel Filter banks are applied to map the linear frequency scale of the FFT output to the Mel Scale. Each Mel filter $H_m[k]$ is a triangular filter defined over the frequency range of interest, and the filter bank energy for the $m^{th}$ Mel band is calculated as Eq. (4) [32].

$$E_m = \sum_{n=0}^{N-1} P[k] \cdot H_m[k] \quad (4)$$

where $H_m[k]$ emphasizes components within the Mel-scaled frequency range. The resulting set of $E_m$ values forms the Mel-Spectrogram, providing a perceptually meaningful time-frequency representation that enhances the performance of audio recognition and classification tasks [31], [32]. The final step is classification. The extracted features are divided into training, validation, and testing sets in this stage. A convolutional neural network (CNN) is employed to develop the classification model. CNNs are a class of deep learning models that excel at recognizing visual patterns, such as images, including spectrogram representations. Their advantages include automatic feature extraction, parameter efficiency, and high accuracy in both classification and detection tasks. Fig. 2 illustrates the basic architecture of a CNN, which consists of several layers, including Convolutional

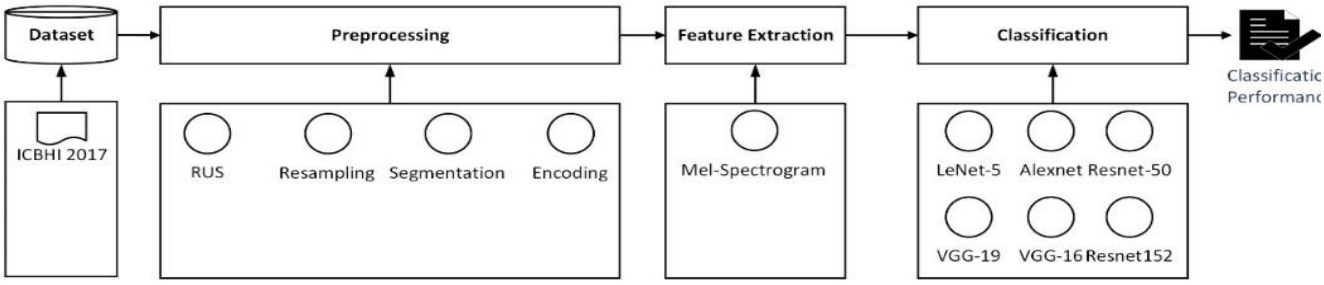



**Fig. 1. Research flow consists of 4 phases, Data collection, preprocessing, feature extraction, and classification.**

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

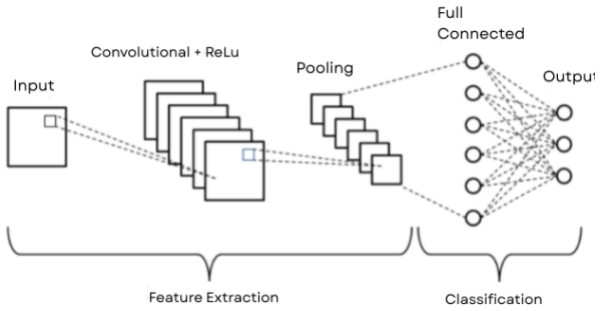



**Fig. 2. The Illustration of basic CNN architecture used for classification and detection task.**

(Conv), Rectified Linear Unit (ReLU), Pooling, and Fully Connected (FC) layers [33], [34]. Mathematically, a convolutional layer that produces the k-th output feature map can be written as Eq. (5) [40].

$$y_{i,j}^{(k)} = \sum_{m=0}^{m-1}\sum_{n=0}^{n-1} X_{i+m,j+n} \cdot W_{m,n}^{(k)} + b^{(k)} \quad (5)$$

Where X is the input feature map, $W^{(k)} \in R^{m \times n}$ is the k-th convolutional kernel, and $b^{(k)}$ its bias. A nonlinear activation (here ReLU) is applied elementwise in Eq. (6) [13]:

$$\sigma(z) = \max(0, z) \quad (6)$$

For pooling, we denote a pooling region P (e.g., 2x2) and stride s; the common operations are max-pooling shown in Eq. (7) [13].

$$O(p,q) = max_{0<i<k_h, 0<j<k_w} I_c(p \cdot s + i, q \cdot s + j) \quad (7)$$

Or average pooling in Eq. (8) [13].

$$O(p,q) = \frac{1}{k_h \cdot k_w}\sum_{i=0}^{k_h-1}\sum_{j=0}^{k_w-1} I(p \cdot s + i, q \cdot s + j) \quad (8)$$

where I is the input feature map and $k_h \times k_w$ is the pooling window size. The fully connected layer that follows the flattened feature maps is written in vector form, as shown at Eq. (9) [13].

$$y_i = f(\sum_{j=1}^{n} w_{i,j} x_j + b_i) \quad (9)$$

where *X* is the input vector to the FC layer, *W* is the weight matrix, *b* is the bias vector, and *f* is the activation function. Finally, for the classification network output using the SoftMax function, where K is the number of classes and $Z_i$ is the logit (the output of the previous layer) for the $i^{th}$ class. The SoftMax function shown in Eq. (10) [13] and as for the loss function during training, this study uses Categorical Cross-Entropy shown in Eq, (11) [39] where *N* is the number of classes, $y_i$ is the true label, and $y\hat{}_i$ is the predicted probability for class *i*, which is suitable for multi-class classification problems.

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (10)$$

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (11)$$

In recent developments, researchers have progressively introduced modifications to the basic CNN architecture to enhance classification performance. Several well-known CNN variants have emerged from such efforts, including LeNet-5, AlexNet, VGG-16, VGG-19, ResNet-50, and ResNet-152 [33], [34].

### 1. LeNet-5

LeNet-5 is a simple and lightweight architecture suitable for classification with a small dataset, such as digit recognition. The architecture illustration can be seen in Fig. 3.(a). LeNet-5 is typically for a greyscale image with a size of 32x32 with a convolutional layer kernel size 5x5 which can be defined as Eq. (12) [21].

$$y_{i,j} = \sum_{m=0}^{4}\sum_{n=0}^{4} X_{i+m,j+n} W_{m,n} + b \quad (12)$$

Average pooling uses a kernel of 2x2 and a stride of 2. As expressed in Eq. (13) [21].

$$O(p,q) = \frac{1}{4}\sum_{i=0}^{1}\sum_{j=0}^{1} I(2p+i, 2q+j) \quad (13)$$

A fully connected layer follows flattened feature maps, as shown at Eq. (9). For the output layer in this research, using SoftMax with four classes, as shown in Eq. (14) [19].

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{4} e^{z_j}} \quad (14)$$

### 2. Alexnet

Introduce ReLu activation, as shown in Eq. (6), that mitigates the vanishing gradient problem, enabling faster convergence and improved training efficiency for large-scale datasets. The architecture illustration can be seen in Fig. 3.(b), the default input shape for AlexNet is 227 x 227 with three color channels or RGB, for a convolutional layer with multiple channels, mathematically can be defined as Eq. (15) [13].

$$y_{i,j,k} = \sum_{c=1}^{C_{in}}\sum_{m=0}^{m-1}\sum_{n=0}^{n-1} X_{i+m,j+n,c} \cdot W_{m,n,c,k} + b_k \quad (15)$$

For the pooling layer, this architecture uses Max-pooling with a kernel of 3 x 3, a stride of 2, and each channel is expressed in Eq. (16) [13].

$$y_{i,j} = max_{(m,n)\in pool} \cdot X_{2i+m,2j+n} \quad (16)$$

Using a flattened fully connected layer input, as shown at Eq. (9) and SoftMax in Eq. (14) as the output layer.

### 3. VGG-16 and VGG-19

VGG-16 and VGG-19 employ uniform 3 x 3 convolutional kernels throughout the network; this consistency facilitates transfer learning across diverse image classification tasks. The VGG-16 architecture illustration can be seen in Fig. 3. (c) and VGG-19 with increased depth that enhances representational capacity for complex visual patterns, with illustration in Fig. 3. (d). With the difference only on depth, the VGG-16 and VGG-19 have the same default input size, which is a 224 x 224 RGB picture, a convolutional layer with multiple channels as shown in Eq. (15), a max-pooling layer with kernel 2 x 2 and stride 2 shown in Eq. (16).

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

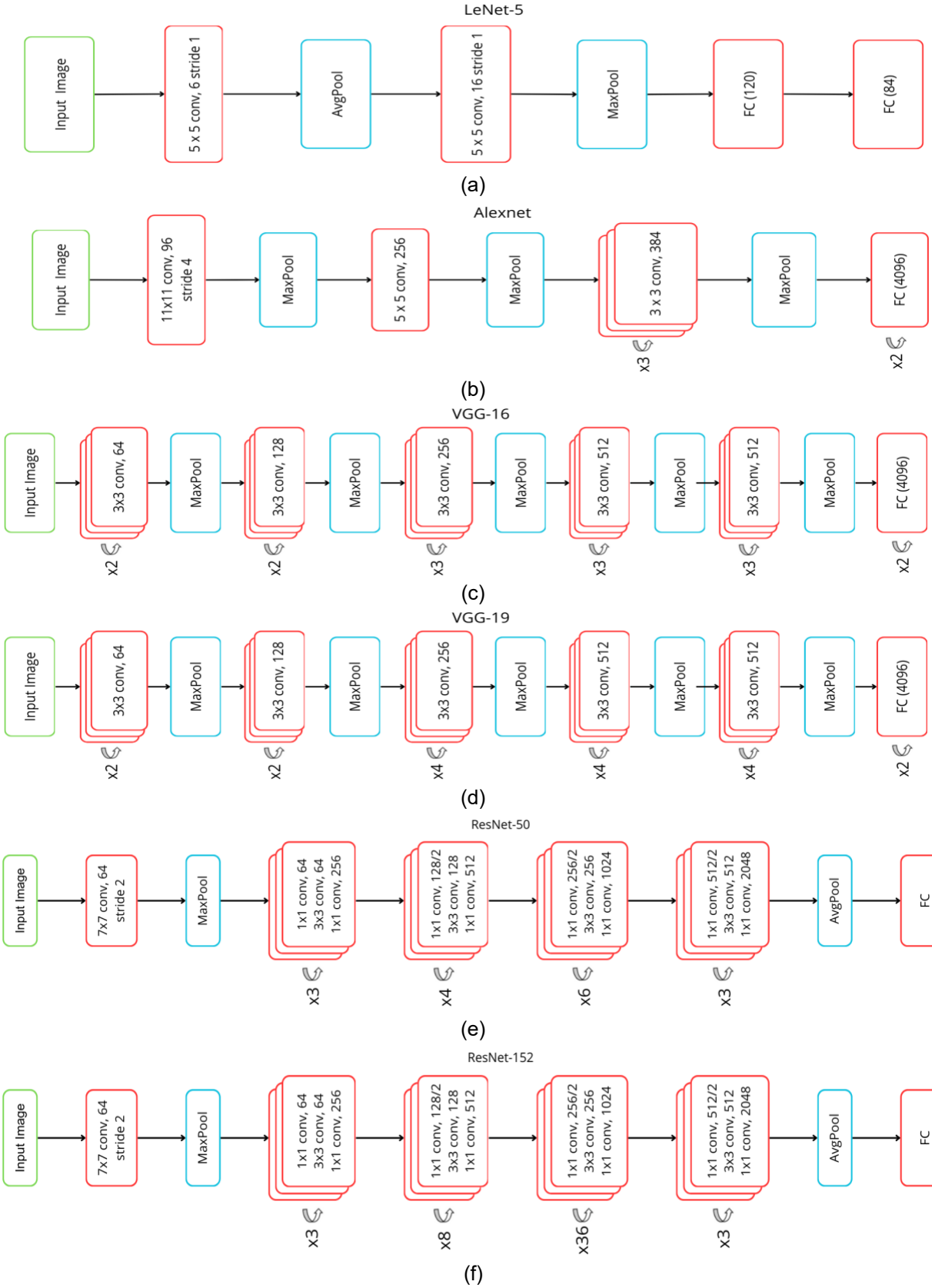



**Fig. 3. Model architecture (a) LeNet-5, (b) Alexnet, (c) VGG-16, (d) VGG-19, (e) ResNet-50, (f) ResNet-152**

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

**Table 1. Summary of CNN architecture variants with differences in depth and composition.**

| Architecture | Input Shape | Layers | Advantage | Ref |
|---|---|---|---|---|
| LeNet-5 | 32×32×1 | Conv → AvgPool → Conv → AvgPool → FC → FC → SoftMax | Simple and lightweight; suitable for small datasets such as digit recognition and grayscale images. | [33], [34] |
| AlexNet | 227×227×3 | Conv → ReLU → MaxPool → Conv → ReLU → MaxPool → Conv ×3 → MaxPool → FC ×2 → SoftMax | Introduced ReLU activation and dropout, which are robust for more complex and larger-scale image data. | [33], [34] |
| VGG-16 | 224×224×3 | Conv ×2 → MaxPool → Conv ×2 → MaxPool → Conv ×3 → MaxPool → Conv ×3 → MaxPool → Conv ×3 → MaxPool → FC ×3 → SoftMax | It uses a consistent convolutional structure and is easily transferable to various image classification tasks. | [33], [34] |
| VGG-19 | 224×224×3 | Conv ×2 → MaxPool → Conv ×2 → MaxPool → Conv ×4 → MaxPool → Conv ×4 → MaxPool → Conv ×4 → MaxPool → FC ×3 → SoftMax | Deeper than VGG-16, capable of capturing more complex visual patterns and high-level features. | [33], [34] |
| ResNet-50 | 224×224×3 | Conv1 () → MaxPool → (Conv ×3 + skip) ×3 → (Conv ×3 + skip) ×4 → (Conv ×3 + skip) ×6 → (Conv ×3 + skip) ×3 → AvgPool → FC → SoftMax | Stable and efficient with residual learning; achieves high accuracy while minimizing overfitting. | [33], [34] |
| ResNet-152 | 224×224×3 | Conv1 → MaxPool → (Conv ×3 + skip) ×3 → (Conv ×3 + skip) ×8 → (Conv ×3 + skip) ×36 → (Conv ×3 + skip) ×3 → AvgPool → FC → SoftMax | A very deep architecture; well-suited for large-scale image classification with high precision. | [33], [34] |

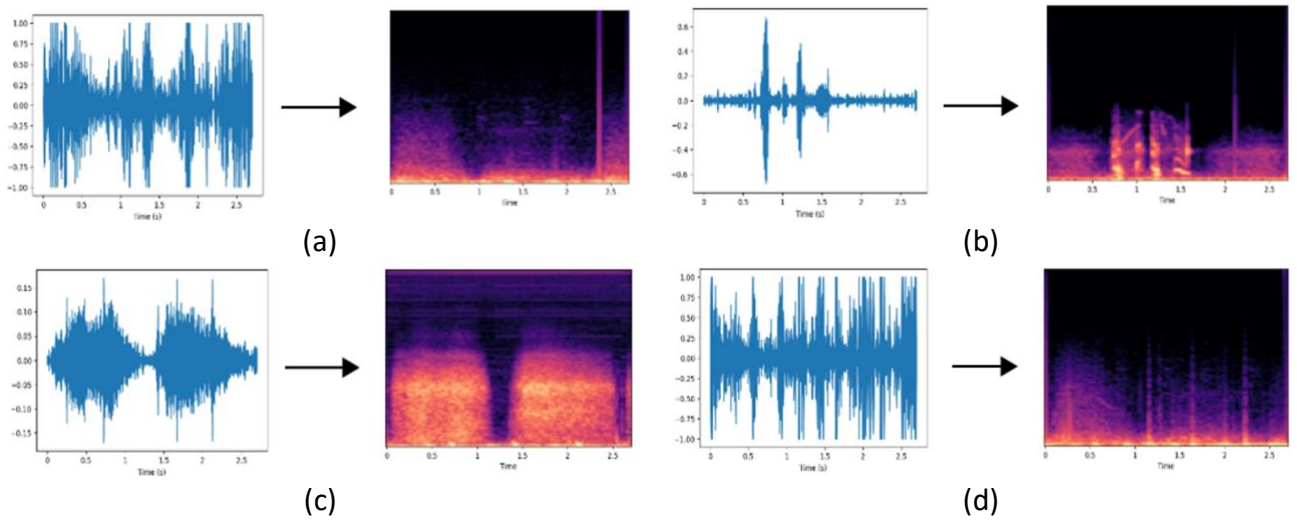



**Fig. 4. Illustration of each class from the audio processed to a Mel Spectrogram picture (a) Crackle, (b) Wheeze, (c) Normal, (d) Crackle & Wheeze**

Fully connected layer that follows flattening before input as shown in Eq. (9) and using ReLu shown at Eq. (6) for first two FC layers and SoftMax as output layer shown in Eq. (14).

### 4. ResNet-50 and ResNet-152

ResNet architecture brought new innovation that introduce residual learning with skip connections, where residual connection ensures stable gradient flow. The identity term in the residual formulation

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

prevents gradient vanishing, achieving high accuracy with fewer parameters than VGG networks while enabling training of deep architectures. A ResNet block learn a residual mapping mathematically shown at Eq. (17) [18].

$$y = F(x, \{W_i\}) + x \quad (17)$$

where X is input (shortcut connection), *F (x, {Wi})* is the residual function (series of conv, BN, ReLU), and the addition (+x) is called a skip connection. For each residual block formula is in Eq. (18) [18].

$$F(x) = W_3 * f(W_2 * f(W_1 * x)) \quad (18)$$

Then the output can be defined as Eq. (19) [18].

$$y = f(F(x) + W_s * x) \quad (19)$$

where $W_s$ is a shortcut projection (1 x 1 conv) if the input and output dimensions differ. The architecture of both ResNet-50 and ResNet-152 begins with an input image of default size 224 x 224 RGB picture, which passes through an initial 7 x 7 convolutional layer with a stride of 2, followed by a 3 x 3 max pooling operation to reduce the spatial dimensions. The key difference between the architecture of ResNet-50 and ResNet-152 is in the number of residual blocks in the four main residual stages. In ResNet-50, the four stages contain 3, 4, 6, and 3 residual blocks, respectively. Each residual block consists of three convolutional layers 1 x 1, 3 x 3, and 1 x 1 filters, along with a shortcut (skip) connection that directly adds the input of the block to its output. With the ResNet-50 architecture, the illustration can be seen in Fig. 3.(e). While ResNet-152 is a deeper version that contains 3, 8, 36, 3 residual blocks across four stages, respectively. The ResNet-152 architecture illustration can be seen in Fig. 3.(f). The structural differences, specifically in the composition and number of layers among these architectures, are summarized in Table 1. For model training, 90% of the data was used for training and 10% for validation. The construction of the models in this study involved experimenting with various hyperparameter settings, as detailed in Table 2. Each model was trained using input shapes compatible with their respective architecture; however, in this study, the input shape was modified to 256 × 256 × 3, deviating from the standard default values. This experimentation with non-default input shapes has not been widely explored in prior state-of-the-art research. Once the best-performing model was identified during training and validation, it was evaluated on the remaining 10% of the dataset for testing. The classification performance of each model was assessed using accuracy as the primary evaluation metric [13], [17], [18]. These parameters are essential for evaluating each model, and hyperparameters such as batch size, number of epochs, and learning rate have a crucial role in the gradient-based learning process. The training process aims to minimize the loss function *J(θ)* by optimizing the model parameter *θ* through gradient-based learning. The parameter update rule follows the gradient descent principle in Eq. (20) [26].

**Table 2. The Experimented Parameter to assess each model in this study.**

| Parameter | Value |
|---|---|
| Batch Size | 64, 128, 256 |
| Epoch | 25, 35, 50, 100 |
| Learning Rate | 0.01, 0.001, 0.0001 |
| Optimizer | Adam, SGD |
| Input Shape | 32, 224, 227, 256 |

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta_t) \quad (20)$$

where *η* denotes the learning rate, controlling the step size of each update, and *∇θJ(θt)* represents the gradient loss function with respect to the model parameters at iteration *t*. The batch size (*B*) influences the stability and variance of the gradient estimation. In mini-batch gradient descent, the update rule becomes as shown in Eq. (21) [26].

$$\theta_{t+1} = \theta_t - \frac{\eta}{B} \sum_{i=1}^{B} \nabla_\theta J_i(\theta_t) \quad (21)$$

where $J_i(\theta_t)$ denotes the loss of $i^{th}$ training example within a batch. Smaller batch sizes introduce higher gradient variance, which may improve generalization but slow convergence, while larger batches produce smoother updates. The number of echoes (*E*) determines the number of complete passes the model makes over the training dataset. Each epoch consists of *N/B* iterations, where *N* is the total number of training samples. Increasing *E* allows the model to refine its parameter estimates, though excessive epochs can lead to overfitting. Through systematic adjustment of these hyperparameters, learning rate, batch size, and epochs, the training process balances convergence speed, stability, and generalization, ultimately improving classification performance.

## III. Result

The result of the preprocessing stage is a balanced dataset, achieved through the implementation of the Random Undersampling (RUS) method [35], [36], where each class, e.g., Crackle, Wheezes, Normal, and both (crackle and wheeze), now contains 500 data. The preprocessing workflow includes several steps: audio resampling to standardize signal amplitude, random undersampling to balance the class distribution, and sample padding to ensure uniform audio duration across all instances [28], [37]. Encoding is then applied to convert categorical labels into a numerical form for further processing [38], [39]. The second stage involves feature extraction using the Mel-Spectrogram method. Parameters configured for the Mel-Spectrogram include frame length and the number of Mel filter banks, which are adjusted based on the input shape to capture the key acoustic characteristics of respiratory sounds effectively.

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

An example of the Mel-Spectrogram resulting from the feature extraction process is presented in Fig. 4. After feature extraction, the dataset was divided into three subsets: 90% for training, 10% for validation, and 10% for testing. The classification model development in this study employed six different CNN-based architectures. For each architecture, combinations of five hyperparameters were varied using predefined values (as detailed in Table 2, resulting in 16 distinct classification models per architecture. The performance metrics of each model are presented in **Table 3**. The best classification accuracy achieved among the models was 75.50%, obtained using 35 epochs, while the remaining parameters followed the configurations shown in **Table 3**. The corresponding default values are listed in the Default Parameter column. In addition, the highest accuracies achieved by individual models using specific parameter adjustments were 73.00% for VGG-19 with 100 epochs, 58.50% for LeNet-5 with a batch size of 256, 71.00% for AlexNet with 100 epochs, 50.00% for ResNet-50 with a learning rate of 0.001, and 57.00% for ResNet-152 using the same parameter learning rate. These findings indicate that variations in key training parameters can significantly affect model performance across different CNN architectures.

## IV. Discussion

Based on the results presented in **Table 3**, several conclusions can be drawn regarding the CNN architectures and parameter configurations that contributed to the best classification performance.

**Table 3. The comparative performance results of the tested parameters and models.**

| Parameter | | Default Parameter | Architecture | F1 Score | Accuracy (%) |
|---|---|---|---|---|---|
| Batch Size | 64 | Learning Rate= 0.0001, Epoch=50, Optimizer=Adam, Shape=227 | VGG-16 | **71,24** | **71,50** |
| | | | VGG-19 | 70.00 | 70,00 |
| | | | LeNet-5 | 53,71 | 54,50 |
| | | | AlexNet | 58,19 | 58,50 |
| | | | ResNet-50 | 0,00 | 0,00 |
| | | | ResNet-152 | 0,00 | 0,00 |
| | 128 | Learning Rate= 0.0001, Epoch=50, Optimizer=Adam, Shape=227 | VGG-16 | **72,97** | **73,50** |
| | | | VGG-19 | 66,93 | 66,50 |
| | | | LeNet-5 | 53,00 | 53,00 |
| | | | AlexNet | 60,29 | 61,50 |
| | | | ResNet-50 | 46,37 | 49,00 |
| | | | ResNet-152 | 35,68 | 40,50 |
| | 256 | Learning Rate= 0.0001, Epoch=50, Optimizer=Adam, Shape=227 | VGG-16 | **71,08** | **72,00** |
| | | | VGG-19 | 69,22 | 69,00 |
| | | | LeNet-5 | 57,88 | 58,50 |
| | | | AlexNet | 51,80 | 53,50 |
| | | | ResNet-50 | 20,94 | 31,00 |
| | | | ResNet-152 | 10,00 | 25,00 |
| Epoch | 25 | Learning Rate= 0.0001, Batch Size=128, Optimizer=Adam, Shape=227 | VGG-16 | 62,74 | 62,50 |
| | | | VGG-19 | **65,55** | **66,50** |
| | | | LeNet-5 | 53,05 | 54,00 |
| | | | AlexNet | 48,69 | 51,00 |
| | | | ResNet-50 | 10,04 | 25,00 |
| | | | ResNet-152 | 10,00 | 25,00 |
| | 35 | Learning Rate= 0.0001, Batch Size=128, Optimizer=Adam, Shape=227 | VGG-16 | **75,31** | **75,50** |
| | | | VGG-19 | 36,31 | 40,00 |
| | | | LeNet-5 | 50,91 | 51,50 |
| | | | AlexNet | 60,70 | 60,50 |
| | | | ResNet-50 | 21,77 | 29,00 |
| | | | ResNet-152 | 19,26 | 29,50 |
| | 50 | Learning Rate= 0.0001, Batch Size=128, Optimizer=Adam, Shape=227 | VGG-16 | **72,97** | **73,50** |
| | | | VGG-19 | 66,93 | 66,50 |
| | | | LeNet-5 | 53,00 | 51,00 |
| | | | AlexNet | 60,29 | 61,50 |
| | | | ResNet-50 | 46,37 | 49,00 |
| | | | ResNet-152 | 35,68 | 40,50 |

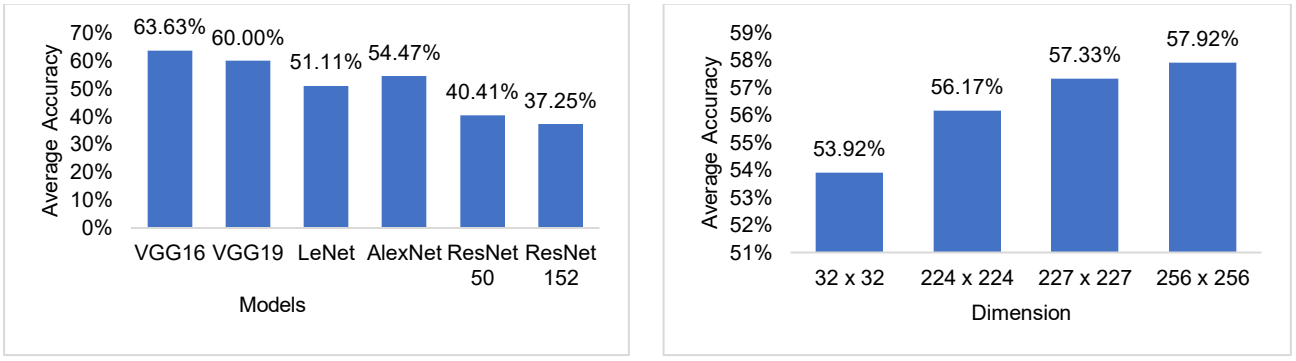
Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

| Parameter | | Default Parameter | Architecture | F1 Score | Accuracy (%) |
|---|---|---|---|---|---|
| | 100 | Learning Rate= 0.0001, Batch Size=128, Optimizer=Adam, Shape=227 | VGG-16 | 69,05 | 69,00 |
| | | | VGG-19 | **72,36** | **73,00** |
| | | | LeNet-5 | 53,93 | 55,00 |
| | | | AlexNet | 70,59 | 71,00 |
| | | | ResNet-50 | 40,73 | 44,00 |
| | | | ResNet-152 | 33,74 | 39,00 |
| Learning Rate | 0,01 | Batch Size=128, Epoch=50, Optimizer=Adam, Shape=227 | VGG-16 | 10,00 | 25,00 |
| | | | VGG-19 | 10,00 | 25,00 |
| | | | LeNet-5 | 10,00 | 25,00 |
| | | | AlexNet | 10,00 | 25,00 |
| | | | ResNet-50 | 46,13 | 48,50 |
| | | | ResNet-152 | **43,12** | **47,00** |
| | 0,001 | Batch Size=128, Epoch=50, Optimizer=Adam, Shape=227 | VGG-16 | 10,00 | 25,00 |
| | | | VGG-19 | 10,00 | 25,00 |
| | | | LeNet-5 | 50,75 | 56,25 |
| | | | AlexNet | 10,00 | 25,00 |
| | | | ResNet-50 | 48,60 | 50,00 |
| | | | ResNet-152 | **56,96** | **57,00** |
| | 0,0001 | Batch Size=128, Epoch=50, Optimizer=Adam, Shape=227 | VGG-16 | **72,97** | **73,50** |
| | | | VGG-19 | 66,93 | 66,50 |
| | | | LeNet-5 | 53,00 | 53,00 |
| | | | AlexNet | 60,29 | 61,50 |
| | | | ResNet-50 | 46,37 | 49,00 |
| | | | ResNet-152 | 35,68 | 40,50 |
| Optimizer | SGD | Batch Size=128, Epoch=50, Learning Rate=0.0001, Shape=227 | VGG-16 | 34,30 | 41,50 |
| | | | VGG-19 | **67,24** | **67,50** |
| | | | LeNet-5 | 23,00 | 30,00 |
| | | | AlexNet | 37,48 | 43,50 |
| | | | ResNet-50 | 35,15 | 38,50 |
| | | | ResNet-152 | 43,03 | 44,50 |
| | Adam | Batch Size=128, Epoch=50, Learning Rate=0.0001, Shape=227 | VGG-16 | **72,97** | **73,50** |
| | | | VGG-19 | 66,93 | 66,50 |
| | | | LeNet-5 | 53,00 | 53,00 |
| | | | AlexNet | 60,29 | 61,50 |
| | | | ResNet-50 | 46,37 | 49,00 |
| | | | ResNet-152 | 35,68 | 40,50 |
| Shape | 32 | Batch Size=128, Epoch=50, Learning Rate=0.0001, Optimizer=Adam | VGG-16 | **63,94** | **63,50** |
| | | | VGG-19 | 59,97 | 59,50 |
| | | | LeNet-5 | 55,03 | 56,00 |
| | | | AlexNet | 56,26 | 57,50 |
| | | | ResNet-50 | 42,69 | 43,50 |
| | | | ResNet-152 | 43,03 | 44,50 |
| | 224 | Batch Size=128, Epoch=50, Learning Rate=0.0001, Optimizer=Adam | VGG-16 | **70,15** | **70,00** |
| | | | VGG-19 | 66,20 | 66,00 |
| | | | LeNet-5 | 56,80 | 57,00 |
| | | | AlexNet | 58,29 | 58,00 |
| | | | ResNet-50 | 38,85 | 42,50 |
| | | | ResNet-152 | 41,11 | 43,50 |
| | 227 | Batch Size=128, Epoch=50, Learning Rate=0.0001, Optimizer=Adam | VGG-16 | **72,97** | **73,50** |
| | | | VGG-19 | 66,93 | 66,50 |
| | | | LeNet-5 | 53,00 | 53,00 |
| | | | AlexNet | 60,29 | 61,50 |
| | | | ResNet-50 | 46,37 | 49,00 |
| | | | ResNet-152 | 35,68 | 40,50 |

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

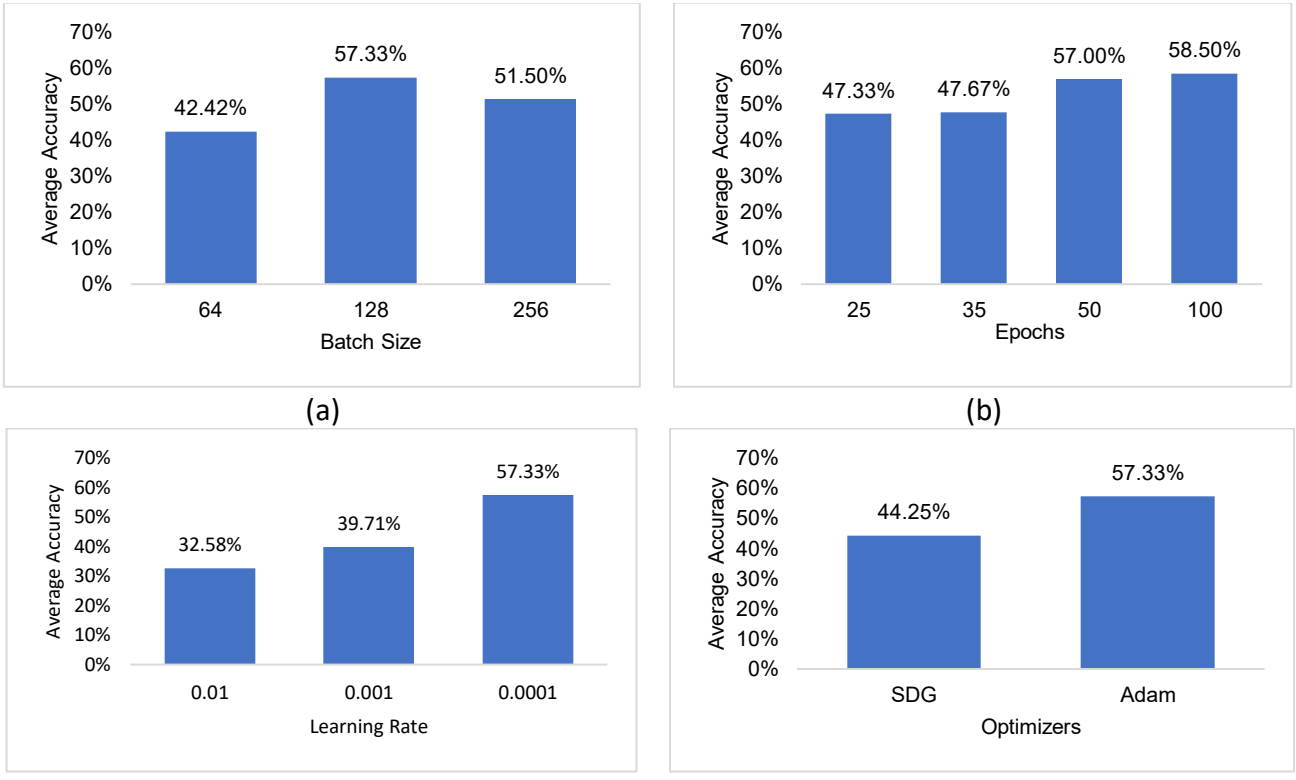
| Parameter | Default Parameter | Architecture | F1 Score | Accuracy (%) |
|---|---|---|---|---|
| 256 | Batch Size=128, Epoch=50, Learning Rate=0.0001, Optimizer=Adam | VGG-16 | **74,58** | **75,00** |
| | | VGG-19 | 66,20 | 66,00 |
| | | LeNet-5 | 56,89 | 57,00 |
| | | AlexNet | 60,42 | 60,50 |
| | | ResNet-50 | 47,79 | 49,50 |
| | | ResNet-152 | 32,40 | 39,50 |

Fig. 5.(a) illustrates the average performance of the six CNN architectures used in this study. Among them, VGG-16 demonstrated the highest overall accuracy, surpassing LeNet, AlexNet, and ResNet. The superior performance of VGG-16 may be attributed to its more complex architecture relative to LeNet, particularly its




(a) (b)

**Fig. 6. Average Performance model and input Dimension (a) Models and (b) Dimensions**




(a) (b)

(c) (d)

**Fig. 5. Average Performance Key Parameters (a) Batch size, (b) Epoch, (c) Learning Rate, and (d) Optimizers**

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

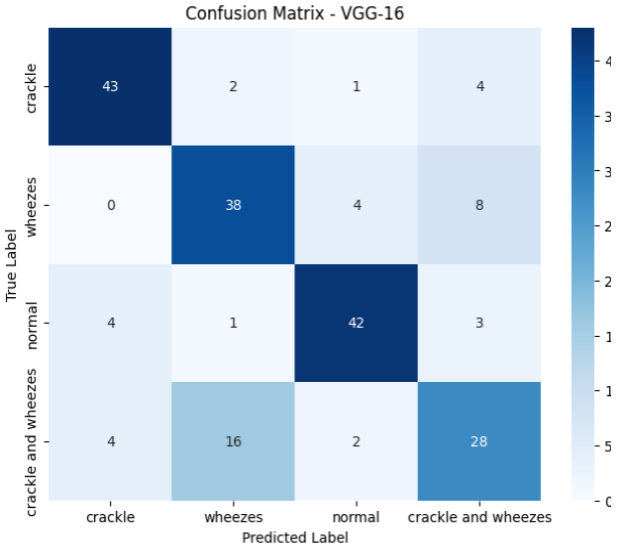



**Fig. 7. Confusion Matrix of our best model for Lung sound Classification**

use of more convolutional layers and ReLU activations, which enable it to capture more detailed and abstract features. VGG-16 also has a more consistent, deeper architecture than AlexNet, as all its convolutional layers use filters of the same size, enhancing its ability to generalize across image classification tasks. Although ResNet is theoretically superior in terms of efficiency, depth, and training stability due to its residual learning framework, in the context of this study, ResNet failed to perform well when applied to spectrogram image data. This result suggests that architectural complexity does not necessarily guarantee better classification performance. A similar pattern was observed between VGG-16 and VGG-19, where, despite VGG-19 being architecturally deeper, VGG-16 consistently achieved better average performance.

Fig. 5(b) shows the impact of input dimensions (input shape) on classification performance across CNN architectures. LeNet, which was originally designed for an input shape of 32×32×1, showed improved performance when tested with larger input dimensions. The influence of input shape on classification accuracy was also evident across other architectures. Although the average performance generally improved with larger input shapes, the highest accuracy for each architecture was achieved with an input shape of 227×227×3. This may indicate that optimal performance is not solely dependent on input size but also on the alignment of other hyperparameter values, such as learning rate, batch size, and training epochs. Fig. 6 presents the average classification performance across variations in several key parameters, including batch size, epoch count, learning rate, and optimizer type. The average results indicate that the best classification performance was achieved with a batch size of 128, 100 epochs, a learning rate of 0.0001, and the Adam optimizer. However, it is essential to note that the best overall classification accuracy, as shown in Table 3, was obtained using a different set of parameter values. We retrained the models using the configuration above to validate the influence of these averaged optimal parameter settings on each CNN architecture. The performance results of these models are presented in the Accuracy (%) column of Table 5. Compared with the maximum accuracies previously achieved by each architecture (shown in the Max Accuracy (%) column), it becomes evident that the models trained using the averaged optimal parameters yielded lower performance across all architectures.

**Table 4. Paired t-Test Results for Model Performance under Different Parameter Settings**

| Parameters | | P Value | Significant |
|---|---|---|---|
| Batch Size | 64 - 128 | 0.1789 | No |
| | 64 - 256 | 0.1995 | No |
| | 128 - 256 | 0.1976 | No |
| Epoch | 25 - 35 | 0.9563 | No |
| | 25 - 50 | 0.0934 | No |
| | 25 - 100 | 0.0161 | Yes |
| Learning Rate | 0,01 | 0.2197 | No |
| | 0,001 | 0.0438 | Yes |
| | 0,0001 | 0.1794 | No |
| Optimizer | SGD - Adam | 0.0704 | No |
| Shape | 32 – 224 | 0.1616 | No |
| | 32 – 227 | 0.1786 | No |
| | 32 - 256 | 0.1238 | No |

Model performance was compared numerically using the accuracy values obtained from each experimental setting. To validate whether the observed differences across parameters were statistically significant, a paired t-test was applied for each parameter, and the results are presented in Table 4.

**Table 5. Performance evaluation of tested models using their best parameter settings.**

| Parameters | Model | Acc. (%) | Max Acc. (%) |
|---|---|---|---|
| Batch Size=128, Epoch=100, Optimizer=Adam, Learning Rate=0.0001, Shape = 256 | LeNet-5 | 55.50 | **58.50** |
| | AlexNet | 64.50 | **71.00** |
| | VGG-16 | 71.00 | **75.50** |
| | VGG-19 | 66.50 | **73.00** |
| | ResNet-50 | **53.00** | 50.00 |
| | ResNet-152 | 46.00 | **57.00** |

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

The results presented in Table 5 indicate that using the best average values of the parameters shown in Fig. 6 did not improve model performance. These findings suggest that selecting optimal parameters for enhancing model performance cannot rely solely on average-based values; instead, a more refined approach is required. One such approach is hyperparameter tuning, which systematically searches for the optimal combination of parameters to improve model performance [40]. Further examination utilizing the confusion matrix, as depicted in Fig. 7, offers a detailed perspective on the model's sorting conduct across diverse respiratory sound categories. The VGG-16 architecture displayed robust capability in recognizing crackle and regular categories, with 43 and 42 correct predictions, respectively. However, the model showed a relatively lower accuracy in detecting crackles and wheezes, where several samples were misclassified as wheezes. This misclassification may occur due to data loss during the audio-frequency processing and to the acoustic similarity between wheezing and combined crackle–wheeze sounds, which share overlapping frequency patterns. Additionally, the wheezes class presented moderate confusion with crackle and wheezes, suggesting that the model has difficulty distinguishing between isolated and co-occurring respiratory events. These results indicate that, although the model effectively recognizes distinct sound patterns, it still faces challenges when dealing with mixed or complex respiratory sounds that exhibit overlapping spectral features.

**Table 6. The performance comparison of our research model against state-of-the-art benchmarks.**

| Year & Ref | Methods | | Accuracy (%) |
|---|---|---|---|
| | Feature Extraction | Architecture | |
| 2022 [17] | Gammatonegram | ResNet-50 | 60.80 |
| 2021 [17] | Gammatonegram | VGG-16 | 67.97 |
| 2022 [18] | Gammatonegram | ResNet-50 | 62.29 |
| 2021 [18] | Gammatonegram | VGG-16 | 62.50 |
| 2021 [18] | Gammatonegram | GoogLeNet | 63.69 |
| 2025 [19] | Mel Spectrogram | MobileNet | 74 |
| Our research | Mel Spectrogram | VGG-16 | **75.5** |

To evaluate the effectiveness of the techniques and parameter configurations proposed in this study, we compared the best-performing model developed in this research with those from state-of-the-art studies that utilized the same dataset. The results of this comparison are presented in Table 6. A comparative study by [18], which employed Gammatonegram feature extraction with ResNet-50, VGG-16, and GoogleLeNet, reported accuracies of 62.29%, 62.50%, and 63.69%, respectively. Another study by [17] also utilized Gammatonegram features and achieved a higher accuracy of 67.97% using VGG-16 by tuning several parameter settings. In contrast, research conducted by [19] using Mel-spectrogram features achieved an accuracy of 74% with MobileNet. However, this study did not apply any preprocessing techniques and was trained on an imbalanced dataset. Our research addresses these limitations and fills this gap by incorporating improved preprocessing, balanced data handling, and optimized architectural configurations to achieve better overall performance.

The performance comparison clearly demonstrates that the combination of techniques used in this study successfully improved the accuracy of lung sound classification models relative to previous research. Although the classification models developed in this study outperformed those in previous state-of-the-art research, their performance remains below the 90%–95% accuracy threshold, indicating considerable room for further methodological improvement. In addition to performance differences, several similarities with previous research were also observed. Consistent with previous studies, our findings show that the results from Mel-Spectrogram feature extraction are superior compared to those state-of-the-art studies using Gammatone feature extraction, and that increasing the network depth does not necessarily lead to better performance on this dataset.

This study used a single audio resampling frequency and a fixed segmentation duration, leaving the impact of varying these parameters on classification performance unexplored. Furthermore, the data balancing technique employed in this research relied solely on Random Undersampling (RUS), a relatively simple method. Thus, future research may benefit from exploring alternative data balancing techniques, potentially leading to further improvements in model performance. Further optimization techniques such as data augmentation and hyperparameter tuning could boost the model's performance. While the suggested model demonstrates promising accuracy in classifying sound, its implementation in an actual clinical environment may face encounter several challenges, including variability in real-world audio conditions, limited dataset diversity, computational limitations, and matters of clarity, regulatory approval, and workflow integration.

## V. Conclusion

The results of this study demonstrate that the combination of preprocessing techniques, audio

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

transformations, and feature extraction methods, classification algorithms, and parameter selection successfully produced a lung sound classification model with an accuracy of 75.5%, utilizing a CNN with the VGG-16 architecture. The optimal model was achieved with the following configuration: Learning Rate = 0.0001, Batch Size = 128, Optimizer = Adam, Input Shape = 227×227×3, and Epochs = 35. Although the achieved performance falls within the "moderate" accuracy range, further improvement is required to reach the "excellent" range of 90%–95%. Therefore, future research should aim to identify more effective techniques to enhance model performance in lung sound classification. Potential directions for future work include optimizing the model using hyperparameter tuning through various strategies such as Grid Search, Random Search, Bayesian Optimization, or Hyperband. This optimization can be further supported by exploring variations in the preprocessing stage, such as different frequency ranges and audio segmentation durations, to better capture relevant acoustic features.

## Acknowledgment

The authors also gratefully acknowledge the collaborative partnership with the Faculty of Transdisciplinary Sciences for Innovation, Kanazawa University, Japan, whose support and cooperation significantly enhanced the scope and quality of this research.

## Funding

This research was financially supported by the Lambung Mangkurat University Research Grant in the fiscal year 2024 under grant number 1374.68/UN8.2/PG/2024.

## Data Availability

The respiratory dataset used in this research uses the dataset from the ICHBI Challenge 2017 and can be downloaded on https://bhichallenge.med.auth.gr/ICBHI_2017_Challenge or on Kaggle at the following link https://www.kaggle.com/datasets/vbookshelf/respiratory-sound-database.

## Author Contribution

Midfai Yabani conceived the research idea, designed the methodology, and contributed to manuscript writing and revisions. Mohammad Reza Faisal developed the analytical framework, carried out data analysis, and contributed to the interpretation of the results. Fatma Indriani implemented the software, developed the models used in this study, and assisted in refining the experimental design. Dodon Turianto Nugrahadi conducted the experiments, collected the data, and prepared the initial draft of the manuscript. Dwi Kartini reviewed the manuscript, provided critical revisions, and contributed to improving the clarity and academic quality of the final paper. Kenji Satou supervised the research and methodology.

## Declarations

### Ethical Approval

The research guide reviewed and ethically approved this manuscript for publication in the Journal.

### Consent for Publication Participants

Consent for publication was given by all participants.

### Competing Interests

The authors declare no competing interests.

## References

[1] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, “A Neural Network-Based Method for Respiratory Sound Analysis and Lung Disease Detection,” *Applied Sciences 2022, Vol. 12, Page 3877*, vol. 12, no. 8, p. 3877, Apr. 2022, doi: 10.3390/APP12083877.

[2] J. Saldanha, S. Chakraborty, S. Patil, K. Kotecha, S. Kumar, and A. Nayyar, “Data augmentation using Variational Autoencoders for improvement of respiratory disease classification,” *PLoS One*, vol. 17, no. 8, p. e0266467, Aug. 2022, doi: 10.1371/JOURNAL.PONE.0266467.

[3] D. M. Huang, J. Huang, K. Qiao, N. S. Zhong, H. Z. Lu, and W. J. Wang, “Deep learning-based lung sound analysis for intelligent stethoscope,” *Military Medical Research 2023 10:1*, vol. 10, no. 1, pp. 1–23, Sep. 2023, doi: 10.1186/S40779-023-00479-3.

[4] L. Pham, H. Phan, R. Palaniappan, A. Mertins, and I. McLoughlin, “CNN-MoE Based Framework for Classification of Respiratory Anomalies and Lung Disease Detection,” *IEEE J Biomed Health Inform*, vol. 25, no. 8, pp. 2938–2947, Aug. 2021, doi: 10.1109/JBHI.2021.3064237.

[5] A. M. Alqudah, S. Qazan, and Y. M. Obeidat, “Deep learning models for detecting respiratory pathologies from raw lung auscultation sounds,” *Soft comput*, vol. 26, no. 24, pp. 13405–13429, Dec. 2022, doi: 10.1007/S00500-022-07499-6/TABLES/4.

[6] Q. Zhang *et al.*, “SPRSound: Open-Source SJTU Paediatric Respiratory Sound Database,” *IEEE Trans Biomed Circuits Syst*, vol. 16, no. 5, pp. 867–881, Oct. 2022, doi: 10.1109/TBCAS.2022.3204910.

[7] F. Wang, X. Yuan, Y. Liu, and C. T. Lam, “LungNeXt: A novel lightweight network utilizing

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

enhanced mel-spectrogram for lung sound classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 8, p. 102200, Oct. 2024, doi: 10.1016/J.JKSUCI.2024.102200.

[8] H. Gulzar, J. Li, A. Manzoor, S. Rehmat, U. Amjad, and H. J. Khan, "DETECTION OF CRACKLES AND WHEEZES IN LUNG SOUND USING TRANSFER LEARNING," *Health Informatics - An International Journal*, vol. 12, no. 2, pp. 01–14, May 2023, doi: 10.5121/HIIJ.2023.12201.

[9] S. Carvalho and E. F. Gomes, "Automatic Classification of Bird Sounds: Using MFCC and Mel Spectrogram Features with Deep Learning," *Vietnam Journal of Computer Science*, vol. 10, no. 1, pp. 39–54, Feb. 2023, doi: 10.1142/S2196888822500300.

[10] P. A. Riadi, M. R. Faisal, D. Kartini, R. A. Nugroho, D. T. Nugrahadi, and D. B. Magfira, "A Comparative Study of Machine Learning Methods for Baby Cry Detection Using MFCC Features," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 1, pp. 73–83, Jan. 2024, doi: 10.35882/JEEEMI.V6I1.350.

[11] S. Kim, J. Y. Baek, and S. P. Lee, "COVID-19 Detection Model with Acoustic Features from Cough Sound and Its Application," *Applied Sciences 2023, Vol. 13, Page 2378*, vol. 13, no. 4, p. 2378, Feb. 2023, doi: 10.3390/APP13042378.

[12] S. Carvalho and E. F. Gomes, "Automatic Identification of Bird Species from Audio," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12672 LNAI, pp. 41–52, 2021, doi: 10.1007/978-3-030-73280-6_4.

[13] R. F. Junaidi *et al.*, "Baby Cry Sound Detection: A Comparison of Mel Spectrogram Image on Convolutional Neural Network Models," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 4, pp. 355–369, Sep. 2024, doi: 10.35882/JEEEMI.V6I4.465.

[14] N. Bacanin *et al.*, "Respiratory Condition Detection Using Audio Analysis and Convolutional Neural Networks Optimized by Modified Metaheuristics," *Axioms 2024, Vol. 13, Page 335*, vol. 13, no. 5, p. 335, May 2024, doi: 10.3390/AXIOMS13050335.

[15] O. H. Anidjar and R. Yozevitch, "Transformer-based language-independent gender recognition in noisy audio environments," *Scientific Reports 2025 15:1*, vol. 15, no. 1, pp. 1–16, Apr. 2025, doi: 10.1038/s41598-025-99011-x.

[16] M. K. Gourisaria, R. Agrawal, M. Sahni, and P. K. Singh, "Comparative analysis of audio classification with MFCC and STFT features using machine learning techniques," *Discover Internet of Things*, vol. 4, no. 1, pp. 1–23, Dec. 2024, doi: 10.1007/S43926-023-00049-Y/TABLES/9.

[17] Z. Neili and K. Sundaraj, "Gammatonegram based Pulmonary Pathologies Classification using Convolutional Neural Networks," *2022 19th IEEE International Multi-Conference on Systems, Signals and Devices, SSD 2022*, pp. 1112–1118, 2022, doi: 10.1109/SSD54932.2022.9955783.

[18] N. Zakaria, F. Mohamed, R. Abdelghani, and K. Sundaraj, "VGG16, ResNet-50, and GoogLeNet Deep Learning Architecture for Breathing Sound Classification: A Comparative Study," *2021 Proceedings of the International Conference on Artificial Intelligence for Cyber Security Systems and Privacy, AI-CSP 2021*, 2021, doi: 10.1109/AI-CSP52968.2021.9671124.

[19] K. V. Suma, D. Koppad, P. Kumar, N. A. Kantikar, and S. Ramesh, "Multi-task Learning for Lung Sound and Lung Disease Classification," *SN Comput Sci*, vol. 6, no. 1, pp. 1–13, Jan. 2025, doi: 10.1007/S42979-024-03506-9/METRICS.

[20] M. Fauzan Nafiz, D. Kartini, M. R. Faisal, F. Indriani, and T. H. Saragih, "Automated Detection of COVID-19 Cough Sound using Mel-Spectrogram Images and Convolutional Neural Network," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 535–548, 2023, doi: 10.26555/jiteki.v9i3.26374.

[21] S. Balasubramaniam, Y. Velmurugan, D. Jaganathan, and S. Dhanasekaran, "A Modified LeNet CNN for Breast Cancer Diagnosis in Ultrasound Images," *Diagnostics 2023, Vol. 13, Page 2746*, vol. 13, no. 17, p. 2746, Aug. 2023, doi: 10.3390/DIAGNOSTICS13172746.

[22] M. S. Jamil, P. N. Gunaratne, and H. Tamura, "A Study on Classification of Faulty Motor Sound Using Convolutional Neural Networks," *Proceedings of International Conference on Artificial Life and Robotics*, pp. 918–922, 2024, doi: 10.5954/ICAROB.2024.GS1-2.

[23] S. Das, S. M. M. Ahsan, M. Rahman, and M. S. Karim, "A Voting Approach for Heart Sounds Classification Using Discrete Wavelet Transform and CNN Architecture," *SN Comput Sci*, vol. 5, no. 2, pp. 1–14, Feb. 2024, doi: 10.1007/S42979-023-02580-9/METRICS.

[24] T. Nguyen and F. Pernkopf, "Lung Sound Classification Using Co-Tuning and Stochastic

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

Normalization," *IEEE Trans Biomed Eng*, vol. 69, no. 9, pp. 2872–2882, Sep. 2022, doi: 10.1109/TBME.2022.3156293.
[25] B. Y. Lu, M. L. Hsueh, and H. D. Wu, "Transmission Perspective on the Mechanism of Coarse and Fine Crackle Sounds," *Archives of Acoustics*, vol. Vol. 46, No. 2, no. 2, pp. 289–300, 2021, doi: 10.24425/AOA.2021.136583.
[26] J. S. Park, K. Kim, J. H. Kim, Y. J. Choi, K. Kim, and D. I. Suh, "A machine learning approach to the development and prospective evaluation of a pediatric lung sound classification model," *Scientific Reports 2023 13:1*, vol. 13, no. 1, pp. 1–10, Jan. 2023, doi: 10.1038/s41598-023-27399-5.
[27] Y. Kim *et al.*, "Respiratory sound classification for crackles, wheezes, and rhonchi in the clinical field using deep learning," *Scientific Reports 2021 11:1*, vol. 11, no. 1, pp. 1–11, Aug. 2021, doi: 10.1038/s41598-021-96724-7.
[28] G. Petmezas *et al.*, "Automated Lung Sound Classification Using a Hybrid CNN-LSTM Network and Focal Loss Function," *Sensors 2022, Vol. 22, Page 1232*, vol. 22, no. 3, p. 1232, Feb. 2022, doi: 10.3390/S22031232.
[29] A. Carlini, C. Bordeau, and M. Ambard, "Auditory localization: a comprehensive practical review," *Front Psychol*, vol. 15, p. 1408073, Jul. 2024, doi: 10.3389/FPSYG.2024.1408073/FULL.
[30] S. Chen, M. Thielk, and T. Q. Gentner, "Auditory Feature-based Perceptual Distance," *bioRxiv*, p. 2024.02.28.582631, Mar. 2024, doi: 10.1101/2024.02.28.582631.
[31] A. Alfaidi, A. Alshahrani, and M. Aljohani, "A Novel Approach to COVID-19 Diagnosis Based on Mel Spectrogram Features and Artificial Intelligence Techniques," *IJCSNS International Journal of Computer Science and Network Security*, vol. 22, no. 9, 2022, doi: 10.22937/IJCSNS.2022.22.9.29.
[32] V. Sareen and K. R. Seeja, "Speech Emotion Recognition using Mel Spectrogram and Convolutional Neural Networks (CNN)," *Procedia Comput Sci*, vol. 258, pp. 3693–3702, Jan. 2025, doi: 10.1016/J.PROCS.2025.04.624.
[33] A. Sebastian, O. Elharrouss, S. Al-Maadeed, and N. Almaadeed, "A Survey on Deep-Learning-Based Diabetic Retinopathy Classification," *Diagnostics 2023, Vol. 13, Page 345*, vol. 13, no. 3, p. 345, Jan. 2023, doi: 10.3390/DIAGNOSTICS13030345.
[34] K. L. Kermanidis, M. Maragoudakis, and M. Krichen, "Convolutional Neural Networks: A Survey," *Computers 2023, Vol. 12, Page 151*, vol. 12, no. 8, p. 151, Jul. 2023, doi: 10.3390/COMPUTERS12080151.
[35] C. Yang, E. A. Fridgeirsson, J. A. Kors, J. M. Reps, and P. R. Rijnbeek, "Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data," *J Big Data*, vol. 11, no. 1, pp. 1–17, Dec. 2024, doi: 10.1186/S40537-023-00857-7/FIGURES/6.
[36] R. M. Pereira, Y. M. G. Costa, and C. N. Silla, "Toward hierarchical classification of imbalanced data using random resampling algorithms," *Inf Sci (N Y)*, vol. 578, pp. 344–363, Nov. 2021, doi: 10.1016/J.INS.2021.07.033.
[37] J. W. Kim, C. Yoon, and H. Y. Jung, "A Military Audio Dataset for Situational Awareness and Surveillance," *Sci Data*, vol. 11, no. 1, pp. 1–10, Dec. 2024, doi: 10.1038/S41597-024-03511-W;SUBJMETA.
[38] M. Hosni, "Encoding Techniques for Handling Categorical Data in Machine Learning-Based Software Development Effort Estimation," *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K - Proceedings*, vol. 1, pp. 460–467, 2023, doi: 10.5220/0012259400003598.
[39] D. Breskuvien˙, G. Dzemyda, D. Breskuvien, and G. Dzemyda, "Categorical Feature Encoding Techniques for Improved Classifier Performance when Dealing with Imbalanced Data of Fraudulent Transactions," *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, vol. 18, no. 3, p. 5433, May 2023, doi: 10.15837/IJCCC.2023.3.5433.
[40] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization," *Heliyon*, vol. 10, no. 5, p. e26586, Mar. 2024, doi: 10.1016/j.heliyon.2024.e26586.

## Author Biography

**Midfai Yabani** is a dedicated student at Lambung Mangkurat University. He has been pursuing his studies in the Department of Computer Science since 2021; his focus is on Data Science, and he has an interest in medical engineering. His research primarily focuses on audio data-driven analysis and Deep Learning, emphasizing the use of innovative, data-driven approaches to solve complex real-world problems. He is passionate about leveraging his

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

advanced technical and analytical skills to drive meaningful technological advancements, ultimately benefiting society. For further details or inquiries about collaboration, please contact him at midfaiybni@gmail.com. He continually strives to expand his knowledge and contribute innovative research solutions for excellence.

**Mohammad Reza Faisal** was born in Banjarmasin. After graduating from high school, he pursued undergraduate studies in the Department of Informatics at Pasundan University in 1995, and later majored in Physics at Bandung Institute of Technology in 1997. After completing his bachelor's program, he gained experience as a training trainer in information technology and software development. Since 2008, he has been a lecturer in computer science at Universitas Lambung Mangkurat, while also pursuing his master's program in Informatics at Bandung Institute of Technology in 2010. In 2015, he furthered his education by pursuing a doctoral degree in Bioinformatics at Kanazawa University, Japan. To this day, he continues to serve as a lecturer in Computer Science at Universitas Lambung Mangkurat. His research interests encompass Data Science, Software Engineering, and Bioinformatics. He can be contacted at email: reza.faisal@ulm.ac.id.

**Fatma Indriani** is a lecturer in the Department of Computer Science at Lambung Mangkurat University, with a strong research interest in Data Science. Before pursuing an academic career, she completed her undergraduate studies in the Department of Informatics at the Bandung Institute of Technology. In 2008, she began her journey as a lecturer at Lambung Mangkurat University, contributing to the field of Computer Science through teaching and research. To further expand her expertise, she pursued a master's degree at Monash University, Australia, which she successfully completed in 2012. Her academic journey continued with a doctorate in Bioinformatics from Kanazawa University, Japan, which she completed in 2022. With a focus on both Data Science and Bioinformatics, she actively engages in research, exploring innovative ways to leverage data-driven technologies for scientific advancement. Her dedication to academia and research allows her to contribute significantly to the development of knowledge in her field, while also mentoring students and collaborating on interdisciplinary projects.

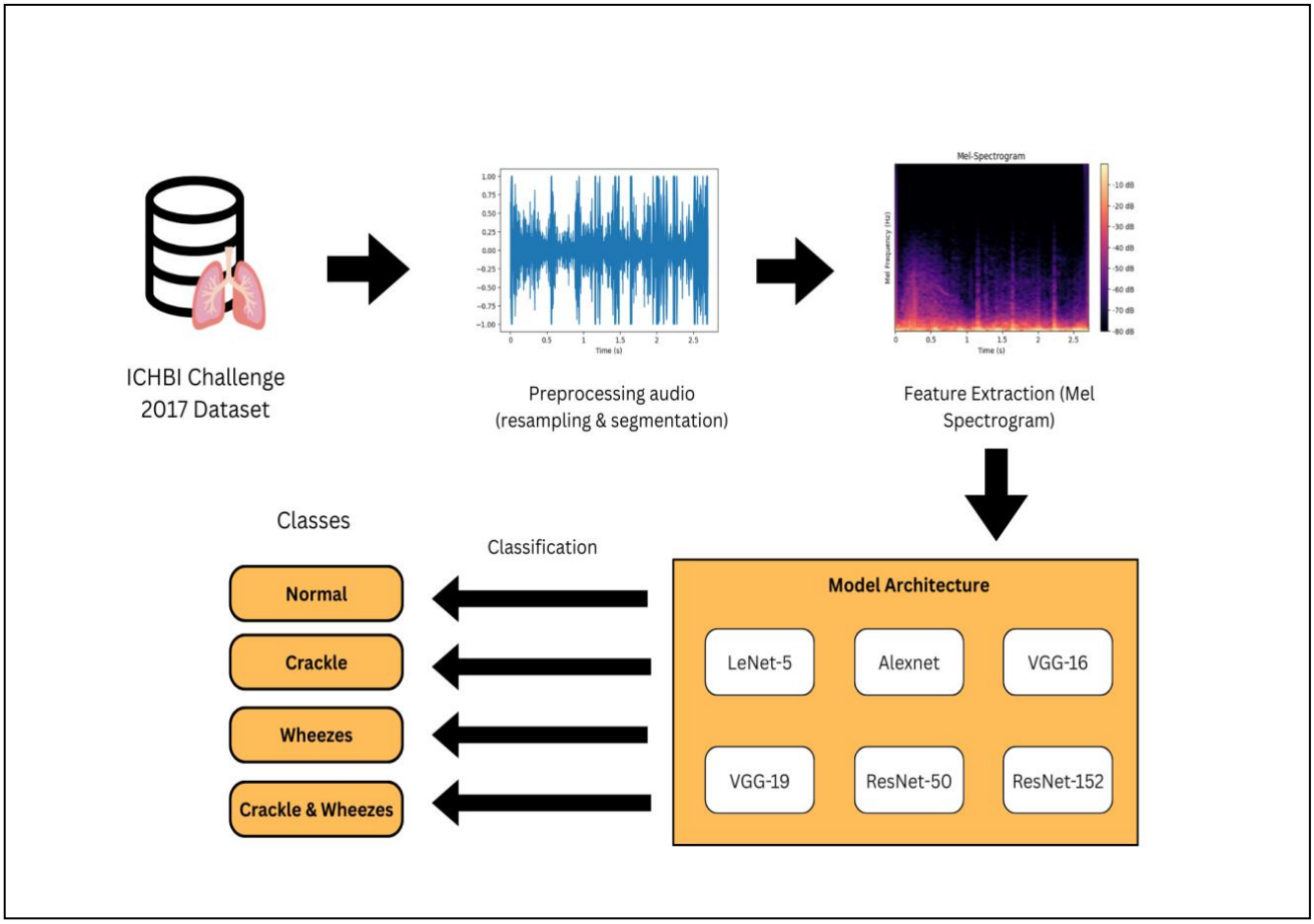
**Dodon Turianto Nugrahadi** is a dedicated academic who contributes to the Department of Computer Science at Lambung Mangkurat University. His scholarly interests converge on the dynamic fields of Data Science and Computer Networking. Having established a strong foundation with a bachelor's degree in informatics engineering from UK Petra, Surabaya, in 2004. He furthered his academic pursuits by obtaining a master's degree in information engineering from Gadjah Mada University, Yogyakarta, in 2009. His current research endeavors delve into the complexities of networks, data science, the Internet of Things (IoT), and network Quality of Service (QoS), demonstrating a commitment to advancing knowledge in these critical areas. He can be contacted at email: dodonturianto@ulm.ac.id.

**Dwi Kartini** earned her Bachelor's and Master's degrees in Computer Science from the Faculty of Computer Science at Putra Indonesia "YPTK" in Padang, Indonesia. She is also a lecturer in the Department of Computer Science, where she teaches a range of courses, including linear algebra, discrete mathematics, and research methods. Her research interests focus on the applications of Artificial Intelligence and Data Mining. Currently, she serves as an assistant professor in the Department of Computer Science, Faculty of Mathematics and Natural Sciences at Lambung Mangkurat University in Banjarbaru, Indonesia. She is the head of the Computer Science study program.

**Kenji Satou** received the B.E., M.E., and D.E. degrees in computer science and communication engineering from Kyushu University in 1987, 1989, and 1996, respectively. He was a research associate at Kyushu University, Fukuoka, Japan (1989-1994), and at the University of Tokyo, Tokyo, Japan (1995-1998). From 1998 to 2007, he was an associate professor at the Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan. He is currently a professor in the Faculty of Transdisciplinary Sciences, the Institute of Philosophy in Interdisciplinary Sciences, Kanazawa University, Ishikawa, Japan. He has a wide variety of interests in bioinformatics, such as Mathematical & Computational Biology, Biochemistry & Molecular Biology, Biotechnology & Applied Microbiology.

Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256

Journal of Electronics, Electromedical Engineering, and Medical Informatics
Homepage: jeeemi.org; Vol. 8, No. 1, January 2026, pp: 168-184 e-ISSN: 2656-8632




Manuscript received 3 October 2025; Revised 10 December 2025; Accepted 20 December 2025; Available online 3 January 2026
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v8i1.1256