Sensors and Materials, Vol. 37, No. 11 (2025) 5141–5162
MYU Tokyo

S & M 4237

Gas Leak Detection in Industrial Air Compressors Using Vision Transformer with Multistage Transfer Learning

Hsuan-Chao Huang,[1] Yuh-Shihng Chang,[2*] and Zheng-Yu Ku[2]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 411030, Taiwan
[2]Department of Information Management, National Chin-Yi University of Technology, No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 411030, Taiwan

(Received September 25, 2025; accepted October 28, 2025)

Keywords: vision transformer, air compressor leak detection, transfer learning, audio feature representation, Mel spectrogram

Recently, machine learning techniques, particularly deep learning models, have been increasingly applied to the analysis, optimization, and prediction of system performance in industrial air compressor leak detection. In this study, we integrated such models for effective gas leak detection based on sensor-derived data, thereby supporting the development of more accurate and efficient factory leak sensing systems to reduce manufacturing costs. We investigated the use of Vision Transformer (ViT) models with transfer learning for gas leak detection in industrial air compressors, based on both audio and visual data. A labeled dataset was created using recordings of compressor sounds under leak and nonleak conditions. Using the intensity stereo localization method with microphone arrays, we estimated the sound intensity and source location. A multistage transfer learning strategy was adopted: ViT models pretrained on ImageNet were adapted with the Environmental Sound Classification (ESC-50) and fine-tuned on real leak data. Among four audio representations, Mel spectrograms achieved the highest accuracy (80%), making them most effective for ViT-based leak detection.

1. Introduction

In recent years, machine learning techniques have been increasingly applied to the analysis, optimization, and prediction of system performance in the field of industrial air compressor leak detection. In this study, we integrated an advanced deep-learning-based model to enhance the effectiveness and accuracy of gas leak detection. With the rapid development of deep learning, audio processing tasks—such as sound classification, speech recognition, and emotion analysis—have achieved significant breakthroughs. Unlike traditional approaches that depend on handcrafted feature extraction, modern deep learning architectures can automatically learn time–frequency representations from audio data, thereby improving the robustness, precision, and adaptability of leak detection systems. In industrial settings, air compressors are widely

*Corresponding author: e-mail: eric_chang@ncut.edu.tw
https://doi.org/10.18494/SAM5950

ISSN 0914-4935

used but often suffer from leakage issues that lead to energy loss and operational inefficiencies. Traditional detection methods—such as visual inspection, stethoscopes, and ultrasound—are either labor-intensive or costly. In addition, with the advancement of hardware computing capabilities, more advanced models such as Convolutional Neural Networks (CNNs) and Transformers have been introduced into audio classification and recognition, further driving the development of this field.[1–4]

In industrial applications, air compressors are critical components used across sectors such as manufacturing, petrochemicals, healthcare, and electronics. However, air leakage in compressor systems poses long-standing challenges, leading to energy waste, reduced production efficiency, increased maintenance costs, and potential safety risks. According to the U.S. Department of Energy, leakage rates in compressed air systems typically range from 20 to 30%, sometimes even higher.[5] Additionally, these systems account for about 10% of total electricity use in the manufacturing sector, making efficient leak detection vital for energy conservation and carbon reduction.[6]

Traditional leak detection methods—such as visual inspection, stethoscopes, soap bubble tests, ultrasonic sensors, and infrared thermography—face limitations in accuracy, cost, and usability. As a result, there is growing interest in AI-powered solutions. Recent advances in audio classification enable AI to identify leak sounds from background noise in real time. While traditional systems rely on handcrafted features such as Mel spectrograms[7] and Mel-frequency cepstral coefficients (MFCCs),[8] modern approaches convert audio into image representations and apply pretrained visual models (e.g., CNNs[9,10] or Vision Transformers) through transfer learning. This strategy has shown promising results in improving detection accuracy while reducing the need for complex training and resources.

In this study, we propose an AI-based audio classification approach using transfer learning with pretrained Vision Transformer (ViT) models.[11,12] Audio signals are transformed into spectrogram images and fine-tuned on datasets such as ESC-50 and real world air compressor leakage data. Multiple feature representations such as raw waveform, log spectrogram, Mel spectrogram, and MFCCs are evaluated. The system visualizes leakage sources using heatmaps, indicating the location and intensity of sound. Results show improved automation and accuracy in leakage detection, demonstrating the potential of ViT-based models in industrial fault diagnosis, including air leak detection, motor fault analysis, and bearing wear monitoring.

2. Data, Materials, and Methods

In this section, we introduce the experimental methods adopted in the development of the gas leak sound detection system presented in this study. The system aims to accurately detect and locate gas leaks in industrial environments by analyzing audio and visual data combined with deep learning techniques. First, we explore in depth the aspects of audio feature extraction, model architecture, and fine-tuning strategies, as well as experimental evaluation metrics.

Subsequently, we compare the performance differences of four different audio feature representations in an audio classification task based on an ImageNet-pretrained model, using ViT as an example. If the AI model classifies the input as a leak sound, the system employs the intensity stereo localization method to locate the sound source. This method estimates sound

intensity by calculating the root mean square (*RMS*) values of each channel in a microphone array and converting them into decibels (dB) to quantify sound strength. The system then determines the spatial location of the sound source based on the energy ratio between multiple channels.

Noted that the ESC-50 dataset used in this study consists of single-channel audio recordings. The term "microphone array" in this paper refers to the conceptual design of the sound source localization module rather than the actual input configuration of the ESC-50 dataset. In the experimental stage, only single-channel signals from the ESC-50 dataset were used for model training and evaluation. The stereo (two-channel) configuration was later implemented in the real-world leak detection prototype system to simulate localization functionality.

To enhance user experience and operational efficiency, we developed an intuitive visual user interface. This interface clearly displays the leak location graphically and provides real-time detection results.

1. If a leak is detected, it displays "Leak Detected"; if no leak is found, it displays "No Leak Detected".
2. Real-time information updates: the video feed, decibel bar chart, and frequency bar chart are also updated in real time, offering users a comprehensive set of visual information.

2.1 Dataset selection

In this study, we adopted the ESC-50 dataset as the experimental platform. ESC-50 is a publicly available and widely used environmental sound classification dataset. It contains 2000 audio recordings of environmental sounds, each 5 s in length, sampled at 44.1 kilohertz (kHz). The dataset is divided into 50 semantic categories, with 40 samples per category. Its diversity and complexity make it an ideal benchmark for validating audio classification methods. All audio files are accessed and processed according to standard formats.

2.2 Audio feature extraction

For each audio sample, we employed four types of feature extraction method: (1) raw audio waveform, (2) log spectrogram, (3) Mel spectrogram, and (4) MFCC. The extracted features are converted into image formats to meet the input requirements of visual models. These feature extraction methods represent a range from low-level to high-level feature representations and are widely applied in the field of audio analysis.[13]

2.2.1 Raw audio waveform

This method preserves the original signal of the audio in the time domain without requiring additional feature engineering, serving as a baseline for comparison. The raw waveform retains all the time-domain features of the audio, including amplitude variations and zero-crossing rate. However, it may have difficulty directly representing frequency-domain characteristics such as harmonic structure and spectral envelope.[14] Figure 1 shows a typical raw audio waveform.

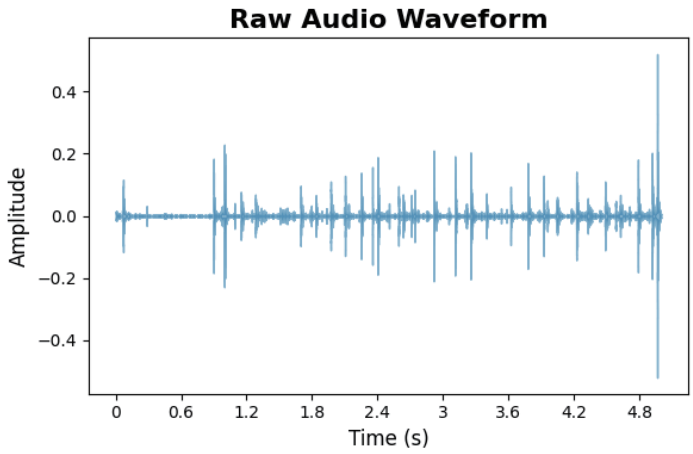


Fig. 1. (Color online) Original audio waveform.

2.2.2 Log spectrogram

By applying a Fourier transform to obtain frequency-domain information, the spectrum is then logarithmically transformed to produce a log spectrogram. The method[15] helps preserve frequency-domain details while reducing the dynamic range of the data, thereby improving recognition performance in certain noisy environments. The log spectrogram retains rich time-frequency features and provides an intuitive visualization of time-varying spectral properties, assisting the model in recognizing harmonic structures, transient variations, and spectral envelopes.[16] A sample log spectrogram is shown in Fig. 2.

2.2.3 Mel spectrogram

To better align with human auditory perception, the Mel spectrogram applies a nonlinear mapping of the spectrum using Mel filters. This approach effectively transforms the original spectral information into a representation consistent with the human auditory system, resulting in superior performance in most speech recognition tasks.[8] The Mel spectrogram simulates the nonlinear frequency sensitivity of human hearing, providing higher frequency resolution in the low-frequency region, which is crucial for identifying low-frequency features in many environmental sounds.[17] Furthermore, the Mel spectrogram has demonstrated excellent performance in recent deep learning studies on audio classification. A sample Mel spectrogram is shown in Fig. 3.

2.2.4 MFCCs

MFCCs are derived by further compressing and extracting features from the Mel spectrogram. This method aims to reduce data redundancy and enhance noise robustness, although it may result in the loss of some detailed information, potentially affecting performance in certain recognition tasks.[18] MFCCs effectively capture the spectral envelope features of

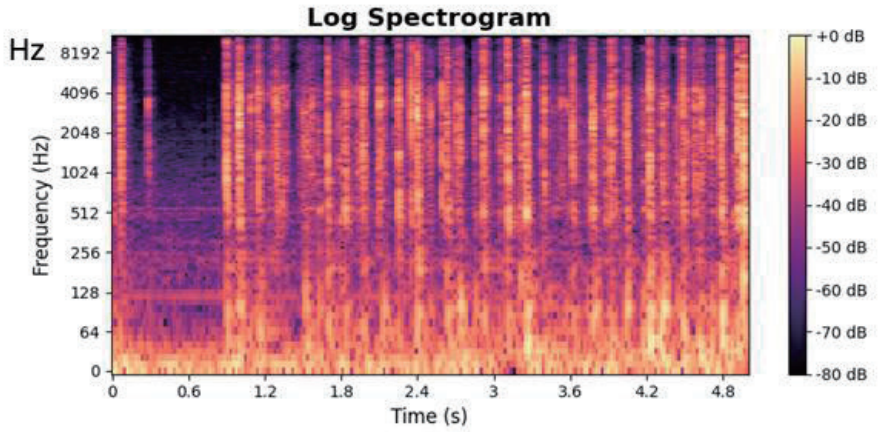


Fig. 2. (Color online) Log spectrum.

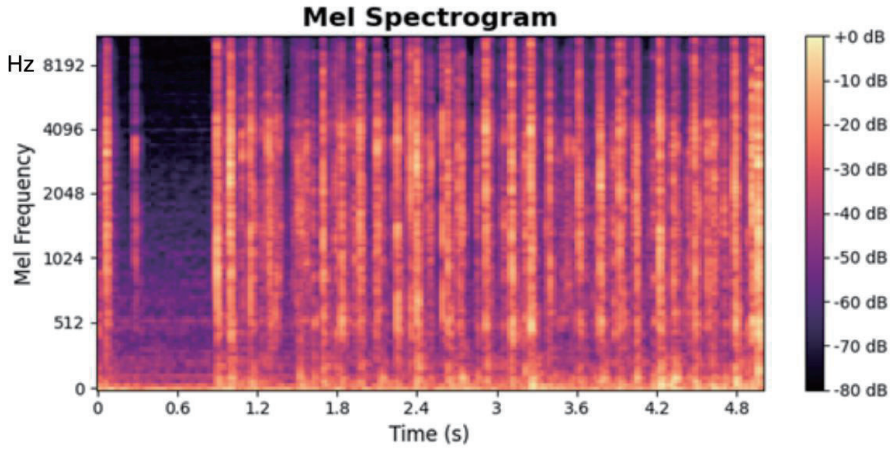


Fig. 3. (Color online) Mel spectrogram.

audio and use Discrete Cosine Transform to decorrelate the features, providing a more compact representation. MFCCs have been widely used in traditional speech recognition and audio classification tasks, especially for identifying features related to the timbre of sound.[19] A sample MFCC is shown in Fig. 4.

2.3 Model architecture and transfer learning strategy

In this study, we adopted the ViT model,[11] which was pretrained on ImageNet,[20] as the base model. The core idea of the ViT model is to divide the input image into several small patches, map them into one-dimensional vectors through linear embedding, add positional encoding, and then input them into a Transformer encoder for feature extraction. Finally, classification is performed through a fully connected layer. Figure 5 illustrates the basic architecture of the ViT model.

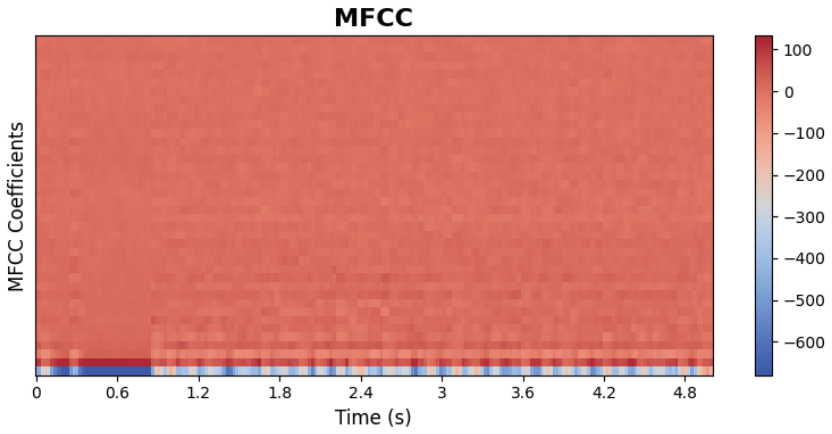


Fig. 4. (Color online) MFCC.

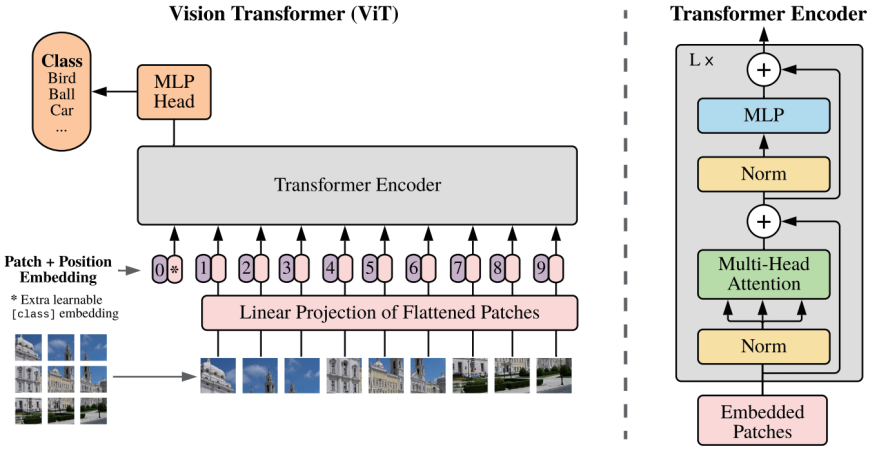


Fig. 5. (Color online) Basic architecture of ViT model.[11]

The ViT model segments the input image into patches, which are then passed through linear embedding and positional encoding before being fed into the Transformer encoder. A fully connected classifier produces the final classification output. In this study, only the final classifier is fine-tuned to adapt to the task requirements of the ESC-50 dataset. The operational principle of the model is shown in Fig. 5. The core structure of the ViT includes the following key components:[21]

1. Patch Embedding: The input image $x \in R^{H\times W\times C}$ is divided into N fixed-size patches, $x_p \in R^{N\times\left(P^2\cdot C\right)}$, where each patch has a resolution of (P, P). The number of patches is given by $N = HW/P^2$. Each patch is then flattened into a vector and linearly projected to obtain patch embedding.[11]
2. Positional Encoding: A learnable positional embedding $E_{pos} \in R^{(N+1)\times D}$ is added to each patch to preserve positional information within the sequence.
3. Class Token: A special classification token (CLS) is prepended to the input sequence, and its final state is used for classification tasks.

4. Transformer Encoder: This consists of multiple Transformer layers, each comprising a multi-head self-attention mechanism and a multi-layer perceptron (MLP) module, with layer normalization and residual connections applied.
5. Classification Head: An MLP layer is added on top of the Transformer output to perform the final classification.

We adopt the ViT-B/16 variants, configured with a patch size of 16 × 16 pixels, a hidden dimension of 768, 12 Transformer layers, 12 attention heads, and an MLP hidden dimension of 3072. To address the audio classification task, a transfer learning strategy is employed in which a pretrained ViT model is adapted using 2000 samples from the ESC-50 dataset. As illustrated in Fig. 6, audio data are first processed to extract features and generate image representations, including raw audio waveforms, log spectrograms, Mel spectrograms, and MFCCs. These image-based inputs are then passed to a ViT model pretrained on ImageNet, with only the final classification layer fine-tuned to produce the ESC-50 classification results.

On the basis of the experimental outcomes, we further developed an air leak detection application system. The system utilizes the best-performing Mel spectrograms as feature

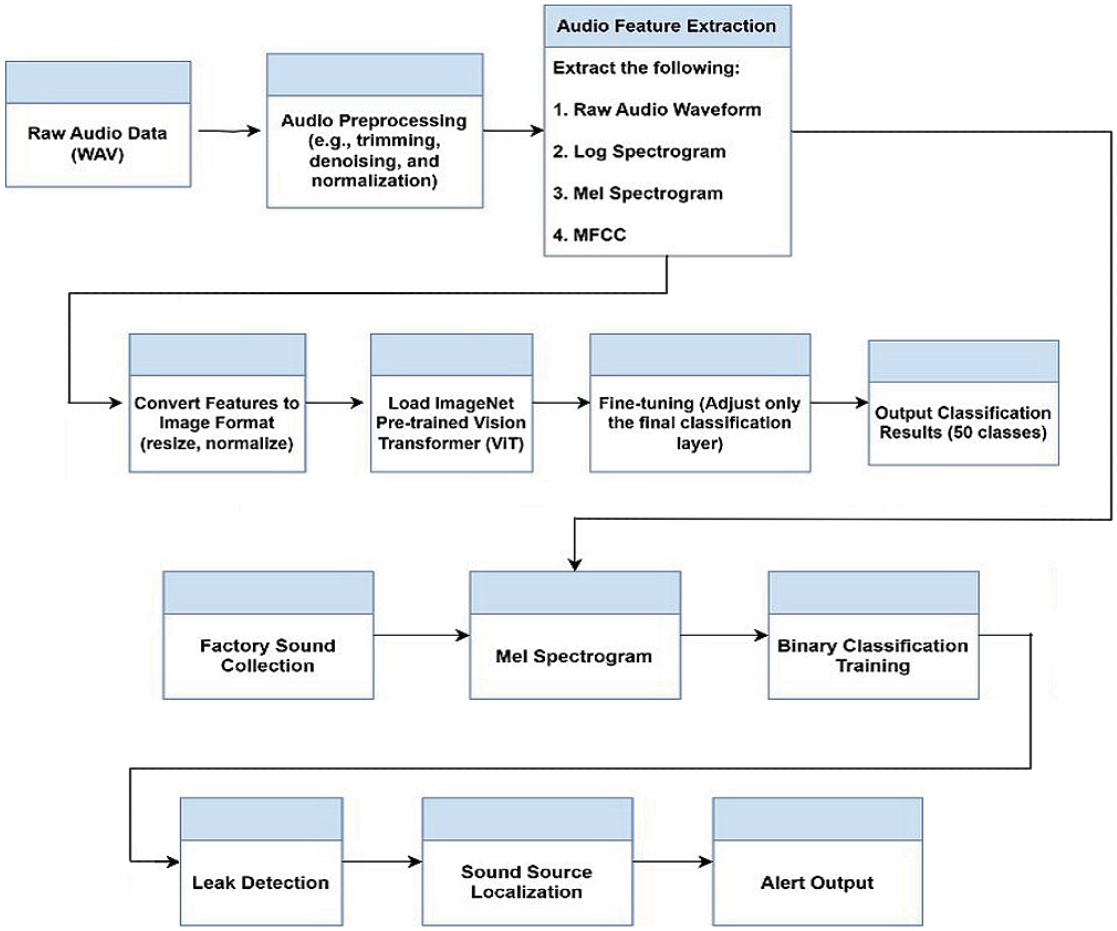


Fig. 6. (Color online) Research flow chart.

representations. On the basis of the initial 2000 audio samples used for training, we further fine-tuned the model with 240 leak samples and 240 normal samples collected from the factory. Each sample was segmented using a two-second time window and converted into Mel spectrograms, which were then used to perform binary classification training with the fine-tuned ViT model. The final system integrates a sound source localization module, enabling real-time leak detection and directional indication, thereby providing a comprehensive industrial air leak detection solution.

2.4 Experimental procedure and evaluation metrics

The experimental procedure is mainly divided into the following steps. First, all audio data from the ESC-50 dataset are preprocessed according to the four aforementioned feature representation methods and converted into image formats compatible with the input requirements of the ViT. Second, models are initialized using identical hyperparameter settings, with the fine-tuning process applied only to the final classification layer. Third, a cross-validation strategy (e.g., fivefold) is employed to ensure the robustness of the experimental results. Finally, each model's performance is recorded and compared in terms of classification accuracy, convergence speed, and computational resource consumption.

The primary evaluation metrics in this study include the following:

1. Classification Accuracy: This measures the model's correct prediction rate on the test set across all categories. It is the most intuitive performance metric, typically defined as the number of correctly predicted samples divided by the total number of samples.[22]
2. Convergence Speed: This evaluates training efficiency by comparing the number of epochs required for the model to converge under different feature representations. Convergence is determined by one or more of the following criteria: no significant improvement in validation accuracy over N consecutive epochs (e.g., 5) and training loss falling below a preset threshold or reaching the maximum number of training epochs.

2.5 Experimental design and parameter optimization

In this study, we adopted a systematic approach to hyperparameter optimization in order to identify the best configuration for each combination of feature representation and model.[23] A coarse grid search was conducted to optimize key hyperparameters, where the learning rate was explored within the range from 1e−5 to 1e−3, the batch size was varied among 16, 32, 64, and 128, the weight decay was tested across 1e−5, 1e−4, and 1e−3, and the dropout rate was adjusted from 0.0 to 0.3. To mitigate overfitting and enhance training efficiency, early stopping is employed with a patience of 10 epochs, where training ceases if validation accuracy shows no improvement, and the parameters yielding the highest validation performance are retained. To ensure the robustness of the experimental results, we adopted a comprehensive cross-validation strategy.[24] Specifically, it follows the original fivefold split of the ESC-50 dataset, using fourfolds for training and onefold for testing in each iteration. To mitigate the effect of randomness, each experiment is repeated three times with different random seeds. Furthermore,

paired t-tests are employed to examine the statistical significance of performance differences between methods, while bootstrap sampling is applied to the test results to compute 95% confidence intervals and evaluate the robustness of the outcomes. The experiments in this study were conducted under the following hardware and software environments, as detailed in Table 1.

Considering the computational cost of large-scale experiments, we adopted the following optimization techniques:[25]

1. Gradient Accumulation: For experiments involving large batch sizes, gradient accumulation is used to reduce GPU memory requirements while maintaining the same effective batch size.
2. Data Preloading and Caching: Multi-process data loading and memory mapping techniques are employed to accelerate the data reading process.
3. Feature Precomputation: Computationally intensive features (e.g., various spectrograms) are precomputed and stored during the data preprocessing stage to avoid redundant computations.

In summary, we introduced the methods adopted in this study for data processing, model selection, and experimental design in this section. Through systematic comparisons of different combinations of audio feature representations and model architectures, we aimed to explore audio classification methods based on vision models and provide effective solutions for environmental sound recognition. In the following sections, we will present the experimental results and further discuss the performance differences and application potentials of various audio feature representations in audio classification tasks.

2.6 Sound source localization module integration

We adopted the intensity stereo localization method for sound source positioning. This method is based on the physical principle that sound intensity decreases proportionally to the square of the distance from the source. When the sound source is biased to one side, the closer microphone receives a stronger signal. The localization procedure is as follows.

1. *RMS* calculation

   For an audio signal $x(n)$ with the sampling rate f_s and the length N, the *RMS* value is calculated as

$$RMS = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x[n]^2}. \tag{1}$$

Table 1
Hardware platform and software environment for this study.

Hardware platform	GPU: NVIDIA GeForce RTX 2060 CPU: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz RAM: 16 GB
Software environment	Operating System: Windows 10 Python Version: 3.8.10 Deep Learning Framework: PyTorch 1.10.0 CUDA Version: 11.3 Audio Processing Libraries: Librosa 0.9.1, torchaudio 0.10.0 Data Analysis Tools: NumPy 1.21.4, Pandas 1.3.4, Scikit-learn 1.0.1 Visualization Tools: Matplotlib 3.5.0, Seaborn 0.11.2

2. Conversion to decibels
   The *RMS* value is converted to dB to provide a more intuitive representation of sound intensity.

$$dB_L = 20 \times \log_{10}(RMS_L) \tag{2}$$

$$dB_R = 20 \times \log_{10}(RMS_R) \tag{3}$$

   When the *RMS* value is 0, a minimum threshold (e.g., 10^{-6}) is set to avoid logarithmic computation errors.
3. Position coordinate calculation
   On the basis of the intensity difference between the left and right channels, the relative position of the sound source is calculated as

$$\Delta dB = dB_L - dB_R. \tag{4}$$

   The position coordinate P is defined as

$$P = \frac{\Delta dB}{dB_{max}} \times 100\%. \tag{5}$$

Here, P ranges from −100 to +100%, where −100% indicates that the sound source is fully to the left, 0% represents a centered position, and +100% indicates that it is fully to the right. dB_{max} represents the predefined maximum intensity difference, typically set to 20 dB. Figure 7 shows the operational flowchart of the gas leak sound detection system developed in this study. Upon system startup, the interface and control components are initialized. Users can choose to press the "Start Detection" or "Stop Detection" button. Once detection is activated, the system simultaneously begins audio recording and activates the camera to capture visual frames. If audio recording fails, an error message will be displayed. If audio recording is successful, the system converts the audio data into a Mel spectrogram and uses a model to determine whether the sound indicates a gas leak.

If a leak sound is detected, the system calculates the decibel difference between the left and right audio channels to perform sound source localization and marks the detected location, displaying a "Leak Detected" message. If no leak is detected, it shows "No Leak Detected." Moreover, the video frame and relevant visual elements (such as decibel and frequency bar charts) are updated in real time, providing users with comprehensive visual feedback. When detection is stopped, the camera is released and visual resources are cleared.

By integrating sound source localization functionality, this system not only detects the presence of air compressor leaks but also provides directional information about the leak source, significantly enhancing its value in practical industrial applications. Maintenance personnel can quickly locate the leak on the basis of the localization information, reducing repair time and further minimizing energy loss.

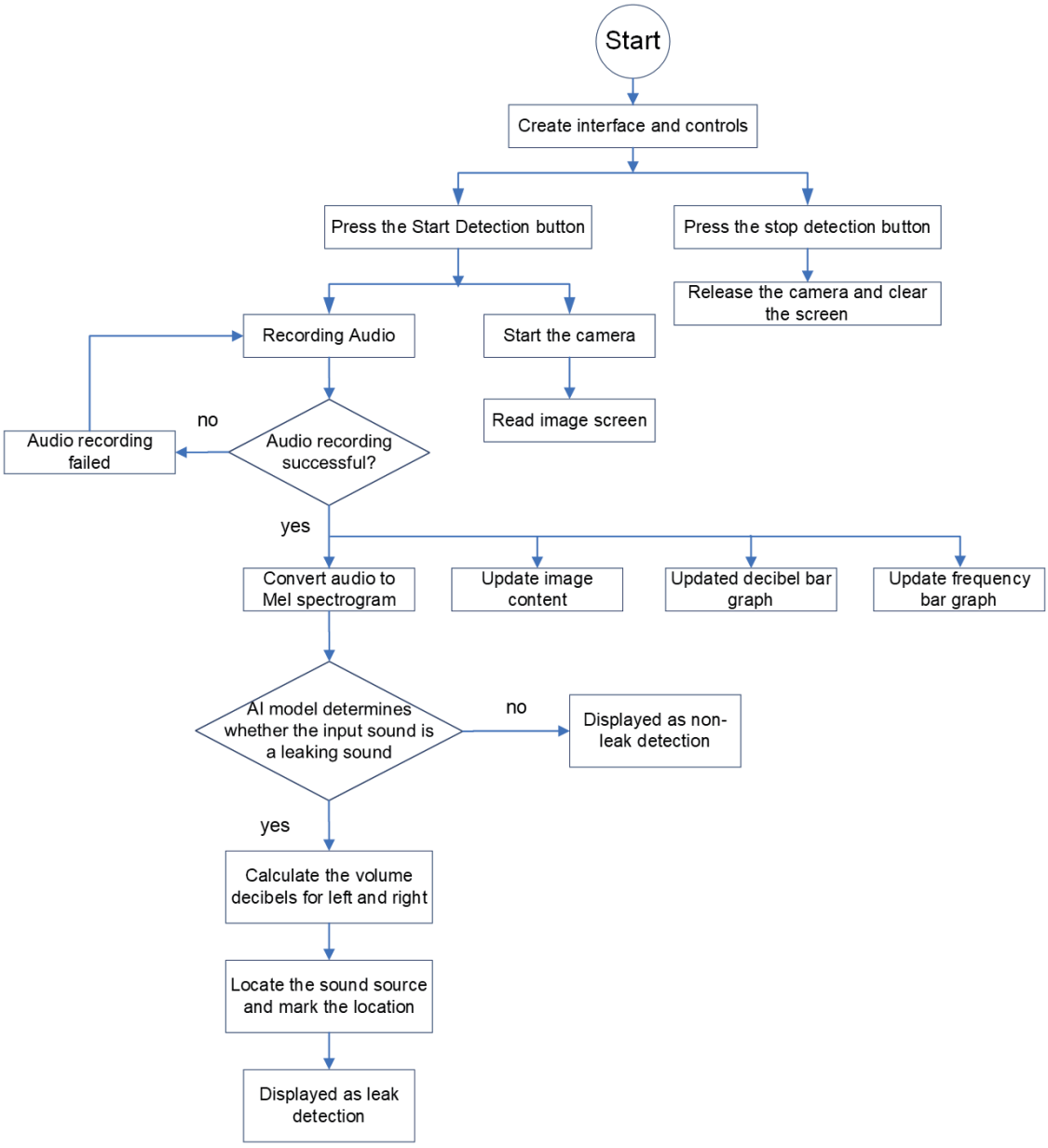


Fig. 7. (Color online) Air leak sound detection system flow chart.

3. Experimental Results and Efficiency Analysis

In this section, we present a comprehensive description of the experimental design, computational environment, and classification performance obtained from different audio feature representations on the ESC-50 dataset. We investigated four primary types of representation, namely, raw audio waveform, log spectrogram, Mel spectrogram, and MFCCs. For classification, a ViT model pretrained on ImageNet is fine-tuned to adapt to the target task. The evaluation emphasizes three key aspects: classification accuracy, model convergence behavior, and computational resource efficiency.

3.1 Dataset configuration

In this study, the ESC-50 environmental sound dataset is used for experimentation. The dataset contains 2000 audio recordings categorized into 50 classes, with 40 samples per class. Each audio file is 5 s in length and sampled at 44.1 kHz. The dataset is predivided into fivefolds to facilitate standardized cross-validation. The ViT model used is based on google/vit-base-patch16-224. The input images are divided into 16 × 16 pixel patches, and the overall input size is 224 × 224 × 3 red–green–blue (RGB) color channels. For fine tuning, the first 10 encoder layers are frozen, while only the last two encoder layers and the classification head are fine-tuned. The classification head has an architecture of 768 → 256 → 10, incorporating a rectified linear unit (ReLU) activation function and a dropout rate of 0.3.

For training settings, the batch size is set to 32 and the initial learning rate is 3×10^{-4}, optimized using the Adaptive Moment Estimation with Weight Decay (AdamW) optimizer with a weight decay of 1×10^{-4}. The learning rate scheduling strategy uses ReduceLROnPlateau, where the learning rate is halved if training stagnates (patience set to 2). The maximum number of training epochs is 30, and an early stopping mechanism is applied if no improvement is observed within three consecutive epochs.

3.2 Comparison of classification accuracy

Among the methods, the Mel spectrogram performed the best in environmental sound classification, achieving an accuracy of 80.0%, significantly outperforming the other three approaches. This result confirms the importance of human auditory characteristics in this task. It also showed the highest convergence speed. The log spectrogram followed, with an accuracy of 71.2%. It retained complete frequency information, highlighting the importance of detailed spectral features compared with MFCCs, and demonstrated higher performance than MFCCs.

The MFCCs showed moderate performance with an accuracy of 66.2%. Although this method performs well in speech recognition, it was less effective in environmental sound classification, likely due to excessive feature compression that caused the loss of critical environmental sound characteristics. Lastly, the raw waveform had the lowest accuracy at 56.2%, owing to the lack of explicit frequency information and the difficulty for pretrained visual models to directly interpret time-domain signals. However, this approach still performed significantly better than random guessing (10%), indicating that the model could learn some temporal patterns to a certain extent.

The experimental results show that different audio feature representation methods exhibit significant differences in performance on classification tasks, as shown in Fig. 8.

3.3 Training process analysis

The training characteristics of the Mel spectrogram demonstrate its superior learning capability. An initial accuracy of 16% indicates the effectiveness of the pretrained features, providing a strong starting point early in training. The training curves exhibit smooth

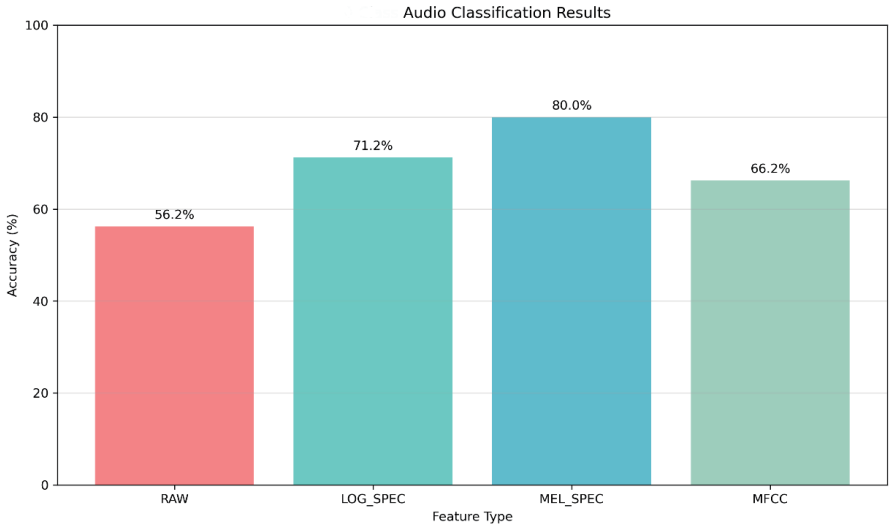


Fig. 8. (Color online) The final test accuracy of four feature representation methods is presented in this study.

progression without signs of overfitting, suggesting that the model is able to learn stably and generalize well. At the 27th epoch, the model achieved its highest performance, triggering the early stopping mechanism, which further confirms the efficiency and stability of the Mel spectrogram for this task. Figure 9 illustrates the training curve of the Mel spectrogram, including the changes in training loss and validation accuracy.

3.4 Acoustic source localization module testing

To build an effective sound recognition model, we collected sound training sample data, covering two main types: air leak sounds and normal sounds. For the air leak sound samples, we created air leak sound sources at the factory site and recorded them within a range of 0–1 step from the sound source. This ensured that the recorded sounds were clear and representative of air leak characteristics. Additionally, we collected normal sounds by walking through various areas of the factory and recording background noises. These included sounds of machinery operation, work-related noise, and general environmental sounds, which will serve as sample data for model training. The categories of collected sound samples are shown in Tables 2 and 3.

The collected normal and leak sound samples were each segmented into 2 s clips. However, since the normal sound was recorded continuously without interruption, some of the normal sound samples did not capture the expected machine operation and factory environment noise. As a result, there are 240 sound samples each for normal and leakage categories in the training dataset. Before training, the program converts the segmented sound samples into Mel spectrograms, which are then used to train the AI model, as shown in Fig. 10.

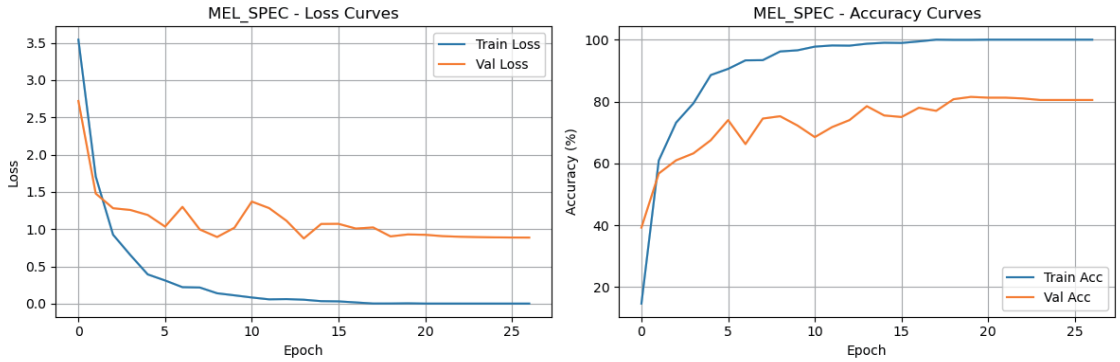


Fig. 9. (Color online) Visualization of Mel spectrogram training process.

Table 2
Leak sound sample category.

Recording distance from sound source (0 steps)	Recording distance from sound source (1 step)	Recording duration
Air compressor fully open_0 steps	Air compressor fully open_1 step	1 min each
Pipeline joint leak_0 steps	Pipeline joint leak_1 step	1 min each
Pipeline joint leak fully open_0 steps	Pipeline joint leak fully open_1 step	1 min each
Pipeline rupture_0 steps	Pipeline rupture_1 step	1 min each

Table 3
Normal sound sample category.

Audio training sample source	Recording duration
Machine operation and ambient sounds within a factory	10.5 min

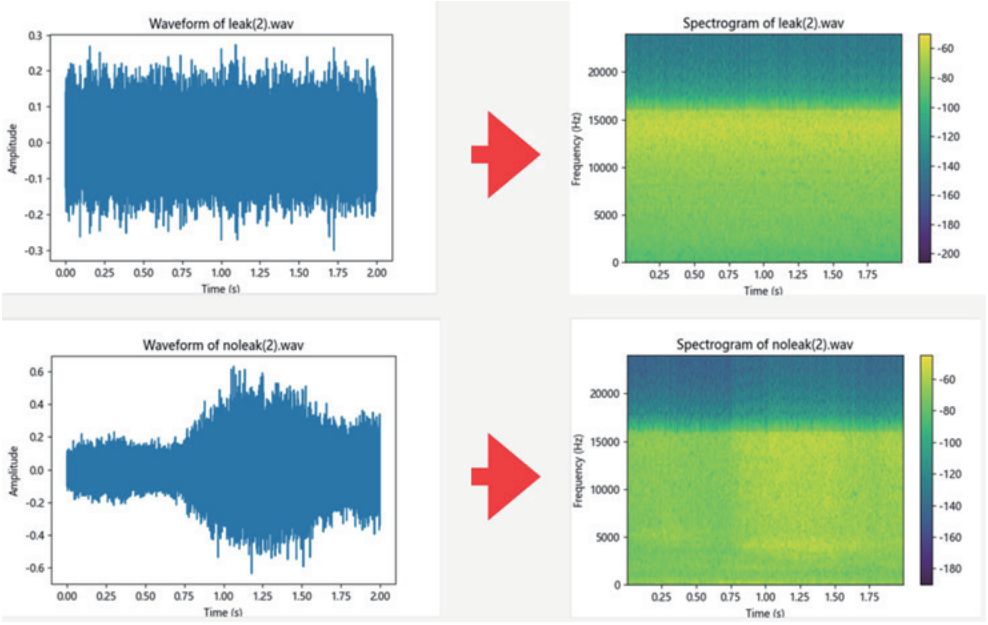


Fig. 10. (Color online) Convert sound samples to Mel spectrograms.

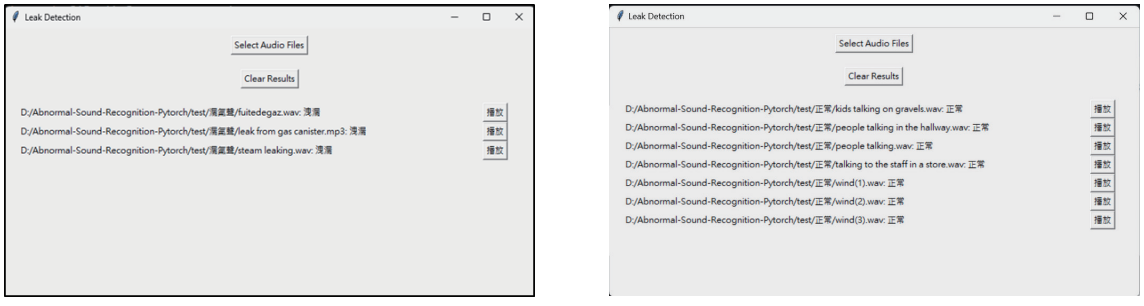
3.5 Unit testing

Each module within the system will undergo individual testing to ensure correct functionality. The scope of testing includes the audio processing, AI model training, and sound source localization display modules. For the audio processing module, testing will focus on the accuracy of audio recording and feature extraction, including signal processing and the correct extraction of Mel spectrogram features. This ensures that the processed data can be correctly fed into either the AI model training module or the sound source localization display module for subsequent processing. For the AI model training module, testing will verify that the training and inference processes function properly, particularly whether the model can accurately distinguish between leak and normal sounds and provide correct prediction results.

Testing for the sound source localization display module will focus on the accuracy of the heatmap and sound source positioning, ensuring that during real-time sound recognition, the system can accurately display the location of the sound source and reflect the corresponding decibel values. The testing methods will include using online audio datasets for validation and comparing the output results of the AI model training module. Additionally, real-world recorded audio tests will be conducted to verify the output results of the sound source localization display module, ensuring that the system meets the expected performance across all aspects, as illustrated in Figs. 11(a). and 11(b).

3.6 System integration testing on actual device

Integration testing is conducted across all modules to evaluate the overall stability and performance of the system, ensuring effective collaboration between modules. The testing scope includes the integration of the audio processing module with the AI model, the integration of the AI model with the sound source localization module, and the end-to-end data flow of the entire system. For the integration testing between the audio processing module and the AI model, the focus is on ensuring that the Mel spectrogram features output by the audio processing module are correctly transmitted to the AI model and used for accurate classification predictions. The integration testing of the AI model with the sound source localization module verifies how the



(a) (b)

Fig. 11. (Color online) Real-world recorded audio detection results are compared with the sound of air leakage. (a) The detection result is the sound of air leakage and (b) the detection result is normal nonleakage sound.

AI model's predictions affect the localization module to ensure that the sound source is correctly displayed on the heat map.

In the overall data flow testing, the entire process, from audio recording and feature extraction to sound recognition and location display, is examined to ensure that each module processes and transmits data accurately. The testing methods include conducting system-wide tests in a simulated environment to assess the collaborative functionality of all components. Particular attention is paid to data flow latency, the accuracy of sound source localization, and overall system stability to ensure high-efficiency and stable operation under various conditions. The sound sources used in this experiment were emitted from a mobile phone, as illustrated in Figs. 12 and 13.

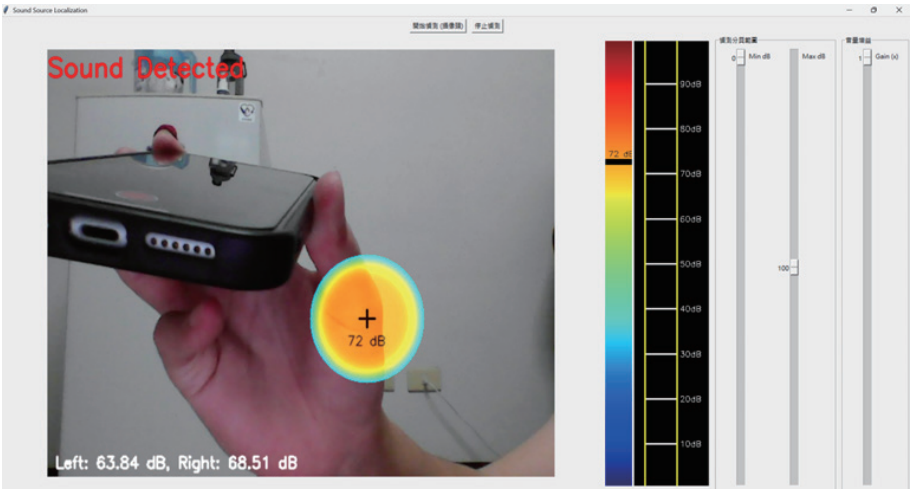


Fig. 12. (Color online) Air compressor leak detection system: sound source localization displays output results.

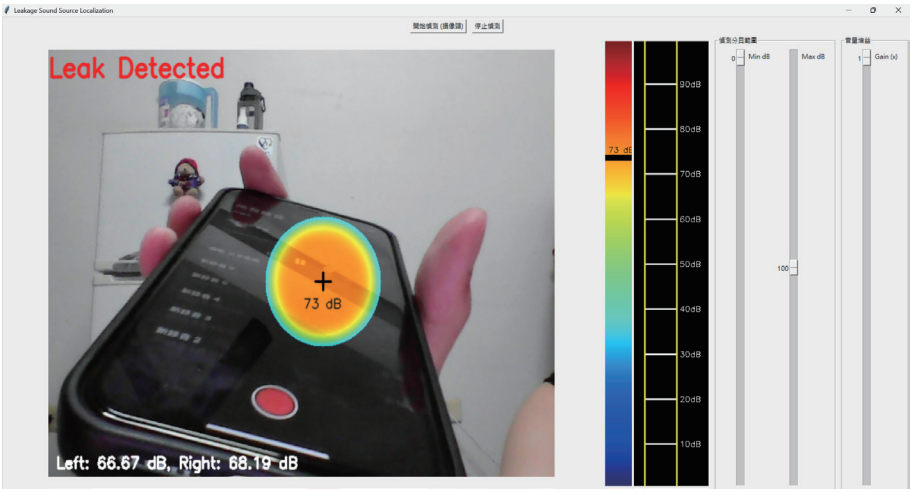


Fig. 13. (Color online) Air compressor leak detection system: Integration results of AI model and sound source location display module.

The gas leak detection system developed in this study consists of both hardware and software components. The hardware configuration includes a dual-channel condenser microphone array (sampling rate: 44.1 kHz) positioned at a fixed distance of 0.5 m from the air compressor to capture stereo sound signals. These signals are processed using a USB audio interface connected to an industrial laptop (Intel i7 CPU, 16 GB of RAM, NVIDIA RTX 2060 GPU). The software system, implemented in Python, integrates three major modules: (1) the audio preprocessing and Mel spectrogram generation module developed using Librosa, (2) the leak classification module employing a fine-tuned ViT model via PyTorch, and (3) the graphical user interface (GUI) developed with PyQt for the real-time visualization of leak detection and localization results.

The system operates in real time, continuously acquiring sound data and transforming them into Mel spectrograms that are input into the ViT model. Once a leak is detected, the intensity stereo localization algorithm calculates the sound source position based on channel energy differences. The GUI then displays a live video feed, decibel variation chart, and frequency spectrum in synchronized time windows. Figures 12 and 13 illustrate the interaction among the hardware components, processing modules, and visualization interface. The system was independently designed and implemented in this study and has not been reported elsewhere.

3.7 Impact of feature representation

The experimental results show that frequency-domain features outperform time-domain features in audio classification tasks. Specifically, all frequency-domain features (such as log spectrograms, Mel spectrograms, and MFCCs) significantly surpassed time-domain features (raw waveforms), with an average accuracy improvement of up to 16.4%.

The fundamental reason why frequency-domain features outperform time-domain features lies in architectural compatibility. The ViT was originally designed for processing two-dimensional visual patterns, and spectrograms naturally exhibit explicit two-dimensional time–frequency structures. Acoustic characteristics such as harmonics and formants form distinctive visual patterns in spectrograms, which align well with the block-wise attention mechanism of ViTs. In contrast, raw waveforms, being one-dimensional temporal signals, lack the spatial structure that ViTs are optimized to process. Even when artificially converted into two-dimensional representations, they fail to produce semantically meaningful spatial relationships, thereby limiting the effectiveness of attention computation and feature extraction.

Furthermore, a ViT pretrained on ImageNet has already acquired capabilities in edge detection and texture recognition. The energy distributions, harmonic structures, and frequency variations in spectrograms closely resemble the texture patterns and edge contours found in natural images. This structural similarity enables the pretrained model to effectively transfer its visual feature recognition abilities to audio analysis tasks. Consequently, even with limited training data, the model achieves strong classification performance, whereas time-domain signals lack such correspondence to visual features.

Among them, the Mel spectrogram achieved an 8.8% higher accuracy than the log spectrogram, demonstrating the importance of simulating human auditory perception in environmental sound classification. Furthermore, although MFCC features are more compact,

their accuracy was 13.8% lower than that of the Mel spectrogram, indicating that excessive feature compression can negatively impact environmental sound classification, particularly when fine-grained information is crucial for accurate classification.

In comparing different feature representations, it is essential to further elaborate on their relevance to transfer learning with ViTs. Spectrogram-based features (e.g., log spectrogram, Mel spectrogram, and MFCCs) encapsulate both temporal and frequency information, enabling ViTs to leverage patch segmentation and self-attention mechanisms to automatically focus on regions with concentrated energy or pronounced frequency variations. For instance, the Mel spectrogram provides higher resolution in the low-frequency range, aligning more closely with human auditory perception, which explains its superior performance in experiments. By contrast, while the log spectrogram preserves the overall energy distribution, it is less sensitive to low-frequency details that are critical in environmental sounds. MFCCs, on the other hand, compress frequency information during feature extraction, which diminishes the model's ability to capture fine-grained acoustic details. Finally, raw waveforms retain the complete temporal signal but lack explicit frequency structures, limiting the capacity of ViTs to establish meaningful time–frequency relationships. Consequently, the frequency and temporal characteristics inherent to each feature representation directly shape how the ViT attention mechanism allocates focus and affects its overall learning effectiveness.

Table 4 presents the spectrogram and feature extraction settings. The log spectrogram is generated using a 1024-point fast fourier transform (FFT) with a hop length of 256 samples. Based on the formula (1+total samples−FFT points)/hop length, this configuration yields 858 time frames, while the frequency dimension is calculated as FFT points/2+1=513, resulting in an original matrix of 858 × 513 size. For the Mel spectrogram, a 2048-point FFT and a hop length of 512 samples are applied. By the same time-frame calculation, 427 frames are obtained, and the frequency dimension is predefined in the referenced study as 128 Mel filters, producing an original dimension of 427 × 128. The MFCCs are then derived from the Mel spectrogram through a discrete cosine transform. While the time dimension remains 427 frames, the feature dimension is expanded to 120 by combining 40 base coefficients with their first- and second-order derivatives, resulting in an original dimension of 427 × 120. These three methods, owing to the differences in window size and hop length, exhibit notable variations in temporal resolution, which subsequently affect the degree of information compression and the performance of ViTs when transformed into 224 × 224 images.

Table 4
Conversion of feature representations into ViT format: time–frequency and feature dimensions.

Feature representation	Original dimension	Final image	Time span per pixel	Frequency/ feature per pixel	ViT patch time	ViT patch frequency/ feature
Raw waveform	220500 samples	224 × 224	22.32 ms	Amplitude quantization levels	357.1 ms	N/A
Log spectrogram	858 × 513	224 × 224	22.24 ms	98.4 Hz	355.8 ms	1575 Hz
Mel spectrogram	427 × 128	224 × 224	22.13 ms	35.7 Hz (Mel scale)	354.1 ms	571 Hz (Mel)
MFCCs	427 × 120	224 × 224	22.13 ms	N/A	354.1 ms	8.6 coefficients

3.8 Effectiveness of transfer learning

In this study, we successfully demonstrated the effectiveness of using vision-based pretrained models in audio tasks, particularly in terms of fast convergence and performance with limited data. Thanks to ImageNet pretraining, the model achieved optimal performance within 30 training epochs, showing the strong learning capabilities of visual models when applied to audio tasks. Even with only 2000 training samples, the model was able to reach an accuracy of 80%, proving its robustness in low-resource scenarios. The ability of visual models to learn textures and patterns transferred effectively to spectrogram analysis, further improving performance in audio classification tasks.

The overall leak detection and localization system demonstrates strong practical potential, particularly in terms of high leak sound recognition accuracy (80%) and precise sound source localization (average error of less than 2%). The system is capable of real-time processing and is suitable for use in industrial environments. However, there are still some limitations: first, it requires stereo recording equipment, which may increase hardware costs; second, performance may degrade in high-noise environments, reducing detection accuracy; and lastly, the system struggles to distinguish multiple simultaneous leak sources, which may pose challenges in certain real-world applications.

In this section, we present a comprehensive evaluation of four types of audio feature representation in the context of Vision-Transformer-based audio classification. The main conclusions are as follows:

1. The Mel spectrogram is the most suitable feature representation, achieving an accuracy of 80%, significantly outperforming other methods.
2. Frequency-domain features clearly outperform time-domain features, with an average performance improvement of more than 16%, confirming the importance of frequency information in environmental sound classification.
3. Visual pretrained models were successfully applied to audio classification, validating the feasibility of cross-modal transfer learning and opening new research directions in audio processing.

These results provide valuable references for future industrial deployment and open new possibilities for cross-modal learning research in the field of audio classification. Future work can explore larger-scale datasets, more complex model architectures, and practical validation in real industrial environments.

Our experimental results indicate that using an ImageNet pretrained model such as ViT for transfer learning can, to some extent, compensate for the mismatch in pretraining audio data within the visual domain. Specifically, the pretrained weights provide a stable initial feature representation, leading to faster convergence during fine tuning. The performance differences observed after fine tuning with various feature representations also highlight the importance of feature engineering in cross-domain transfer. Furthermore, for embedded or resource-constrained applications, selecting appropriate audio features, such as MFCCs or Mel spectrograms, can balance performance and computational efficiency, thereby improving the practical viability of the application.

In terms of practical contributions, the system simplifies operational procedures through an intuitive visual interface, allowing on-site personnel to operate the system without specialized acoustic knowledge and consequently lowering training costs. The automated detection system also enhances detection efficiency through continuous operation, enabling prompt leak identification and the prevention of energy waste. According to the literature, timely leak repairs can reduce compressed air system energy consumption by 20–30%.

4. Conclusions

In comparison with previous studies, most conventional air leak detection systems rely on ultrasonic sensors or CNN-based models trained on handcrafted audio features such as MFCCs or log-mel spectrograms. These methods often require extensive data collection and struggle with generalization in noisy industrial environments. In contrast, in this study, we introduced a ViT-based framework that integrates multistage transfer learning—from ImageNet to ESC-50 and finally to real-world industrial data. This hierarchical adaptation enables effective knowledge transfer from large-scale visual domains to audio-based tasks, substantially improving detection accuracy (80%) even with limited training data. Furthermore, we developed a complete real-time leak detection system combining sound localization and a GUI, demonstrating both academic and practical novelty. To our knowledge, few previous works have achieved such cross-modal integration of visual transformer architectures and industrial acoustic sensing for gas leak detection applications.

ViT is a deep learning model that is often considered a “black box” owing to the difficulty in interpreting its decision-making processes. It operates by dividing an input image into small patches and then using a self-attention mechanism to learn the relationships between these image patches. The model’s parameters are progressively adjusted during training to optimize the output. A significant advantage of ViT regarding interpretability lies in the inherent explainability of its self-attention mechanism. By visualizing the attention weights, we can observe which areas of an image the model tends to focus on when making a prediction. Furthermore, we can analyze the behavior of different attention heads at various layers to understand whether they specialize in focusing on edges, textures, or specific object regions.

In this study, we demonstrated that a ViT model, pretrained on ImageNet, can be effectively applied to audio classification tasks. By converting audio signals into a visual representation, we successfully leveraged the powerful feature extraction capabilities of a vision model. This research goes beyond theoretical validation by developing a complete, practical system that includes an intuitive graphical interface, real-time spectral analysis, and a leakage alert function. The system’s modular design makes it easy to deploy and maintain, laying the foundation for practical industrial applications. The findings of this study can be applied to enhance sensor-based systems for industry gas air leak detection. The integration of sensor data and analytical techniques contributes to the advancement of smart monitoring and diagnostic technologies.

In this study, we demonstrate the successful application of cross-modal transfer learning by effectively adapting a ViT model, pretrained on ImageNet, for audio classification. This was accomplished by converting audio signals into visual representations, thereby leveraging the

powerful feature extraction capabilities of visual models. Beyond theoretical validation, we developed a complete, real-world system with a user-friendly graphical interface. This system includes real-time spectrogram analysis and a leak alert function. Its modular design ensures ease of deployment and maintenance, paving the way for practical industrial applications.

Future development will focus on incorporating digital filters, such as band-pass filters, to address background noise and optimize detection for the specific frequency range of air compressor leak sounds. We will also expand the dataset and refine model adjustment strategies to improve audio classification performance. Additionally, we plan to explore multi-feature fusion methods to enhance the system's applicability and generalization capabilities.

References

1 S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson: arXiv:1609.09430 (2016) https://arxiv.org/abs/1609.09430 (accessed 21 July 2025).
2 Q. Kong, C. Yu, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley: arXiv:2007.07966 (2020) https://arxiv.org/abs/2007.07966 (accessed 21 July 2025).
3 J. Gong, S. Huang, and H. Geng: arXiv:2007.11154 (2020) https://arxiv.org/abs/2007.11154 (accessed 21 July 2025).
4 Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley: arXiv:1912.10211 (2019) https://arxiv.org/abs/1912.10211 (accessed 21 July 2025).
5 Compressed Air Best Practices: http://www.airbestpractices.com/system-assessments/leaks/protect-profits-compressed-air-leakage-best-practices (accessed 21 July 2025).
6 U.S. Department of Energy: Improving Compressed Air System Performance: A Sourcebook for Industry, 2nd ed. (Office of Industrial Technologies, Energy Efficiency and Renewable Energy, Washington, DC, 2003) Chap. 1 https://www.energy.gov/eere/amo/compressed-air-systems (accessed April 2025).
7 P. Mermelstein: Pattern Recognition and Artificial Intelligence, C.H. Chen, Ed. (Academic Press, New York, 1976) Vol. 116, 374–388.
8 S. Davis and P. Mermelstein: IEEE Trans. Acoust. Speech Signal Process. **28** (1980) 357. https://doi.org/10.1109/TASSP.1980.1163420
9 K. Palanisamy, D. Singhania, and A. Yao: arXiv:2007.11154 (2020). https://doi.org/10.48550/arXiv.2007.11154
10 J. Lee, H. Kim, and J. Choi: Appl. Sci. **11** (2021) 3043. https://doi.org/10.3390/app11073043
11 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby: arXiv preprint arXiv:2010.11929 (2020). https://arxiv.org/pdf/2010.11929/1000
12 A. A. González-Hernández, M. Moreno-Álvarez, R. D. Rivera-Rangel, and R. J. Romero-Troncoso: Information **15** (2024) 751.
13 S. Dieleman and B. Schrauwen: 2014 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Florence, Italy, 4–9 May 2014 (2014) 6964–6968.
14 S. Abdoli, P. Cardinal, and A.L. Koerich: Signal Process. Image Commun. **73** (2019) 16.
15 M. Huzaifah: arXiv:1706.07156 (2017).
16 B. Boashash: Time-Frequency Signal Analysis and Processing: A Comprehensive Reference (Academic Press, Oxford, UK, 2015) 2nd ed.
17 T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals: Proc. 16th Annu. Conf. Int. Speech Commun. Assoc. (Interspeech 2015), Dresden, Germany, 6–10 September 2015 (2015) 1–5.
18 F. Eyben, S. Böck, B. Schuller, and A. Graves: Proc. 11th Int. Soc. Music Inf. Retr. Conf. (ISMIR), Utrecht, The Netherlands, 9–13 August 2010 (2010) 589–594.
19 A. Mesaros, T. Heittola, and T. Virtanen: Appl. Sci. **6** (2016) 162.
20 J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei: 2009 IEEE Conf. Comput. Vis. Pattern Recognit., Miami, FL, USA, 20–25 June 2009 (2009) 248–255.
21 K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, et al.: IEEE Trans. Pattern Anal. Mach. Intell. **45** (2022) 87. https://doi.org/10.1109/TPAMI.2022.3152247

22 M. Grandini, E. Bagli, and G. Visani: arXiv:2006.09212 (2020).
23 M. Feurer and F. Hutter: Automated Machine Learning: Methods, Systems, Challenges, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. (Springer, Cham, 2019) pp. 3–33. https://doi.org/10.1007/978-3-030-05318-5_1
24 R. Kohavi: Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI), Montreal, QC, Canada, 20–25 August 1995 (1995) Vol. 2, 1137–1145 https://www.ijcai.org/Proceedings/95-2/Papers/016.pdf (accessed 21 July 2025).
25 P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, and H. Wu: arXiv:1710.03740 (2017).