Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/48/20241648

Application of deep learning in noise detection of electromechanical products

Guoli Zhao[1], Weini Weng[1,2,3]

[1]Bosch Power Tools (China) Co., Ltd., Hangzhou, 310052, China

[2]weini.weng@bosch.com
[3]Corresponding author: Weini Weng, Source: Bosch Power Tools Quality Digitalization Project, Email: weini.weng@bosch.com

Abstract. In recent years, machine learning (ML) techniques have been widely integrated into manufacturing processes. This study explores supervised learning-based machine learning models designed for noise detection. The approach involves feature engineering, extracting relevant features from signals to provide meaningful inputs for machine learning models. Various algorithms, including convolutional neural networks (CNN) and other deep learning architectures, are employed to capture complex dependencies within the data. Training processes involve optimization techniques to enhance model performance and generalizability. Performance evaluation on benchmark datasets compares the proposed model with existing noise detection methods. Results demonstrate the outstanding accuracy and robustness of the generated machine learning model in distinguishing noise from clear signals. Additionally, the study explores the transferability of these models to new scenarios, highlighting their adaptability in practical applications. Findings from this research contribute to advancing noise detection methods, showcasing the effectiveness of machine learning in addressing this crucial challenge. The proposed solution is cost-effective, structurally simple, and holds vast potential for widespread implementation in factory settings in the future.

Keywords: Deep Learning, Noise Detection, Sound Collection, Machine Learning

1. Introduction

Noise is a common phenomenon in equipment usage, often overlooked by factory workers as it doesn't directly cause machinery failures. Consequently, the hazards posed by noise remain highly concealed. Unless remarkably sharp and discomforting, noise tends to go unnoticed, and measures are only taken if it causes a considerable sense of discomfort. People often work in environments with persistent noise, impacting sleep and rest if exceeding 50 decibels in daily life. However, in industrial production settings, inspecting the noise of electromechanical products is crucial for ensuring product quality. Noise sources in electromechanical products, such as friction between mechanical components during machine operation or vibrations from motors, need verification during equipment shipment. This ensures proper functioning, avoiding early equipment failures caused by excessive noise or vibrations. In discrete manufacturing, manual noise inspection is prevalent, exposing workers to potential hearing impairments and significant health risks. Leveraging big data, machine learning, and audio processing techniques, we developed a machine learning-based method for equipment noise detection. Different from

DOI: 10.54254/2755-2721/48/20241648

traditional methods, its application in electromechanical product assembly lines enhances noise detection accuracy and efficiency, safeguarding employees' hearing health while improving production line efficiency.

2. Literature Review

Various methods exist for detecting noise in electromechanical products. Firstly, subjective assessment relies on personnel using their ears to detect noise. Experienced workers can judge equipment status based on emitted noise, often pinpointing faulty components. However, this method remains qualitative, lacking standardized measurement of noise sources. Near-field measurement involves using sound level meters to scan machinery closely, analyzing abnormal sound positions by decibel level. This method is prone to environmental and instrument precision disturbances, yielding less accurate results suitable for preliminary, low-frequency, and simple noise checks. Additionally, spectrum analysis involves measuring motor vibrations using vibration measurement instruments (e.g., accelerometers, vibration sensors) to evaluate noise levels based on spectral characteristics. This approach is widely applied in machines with bearings, studying noise frequency peaks to identify main noise components. For instance, Wang Fei [1] simulated bearing noise detection using MATLAB software, analyzing voltage signals collected by a sound sensor through waveform signal processing. In metallurgical machinery noise research, Yu Caiwen [7] proposed effective noise inspection methods, converting vibration and noise signals into spectra to intuitively display energy distributions across different frequency components, aiding in identifying main sources of vibration and noise.

Some researchers extend noise detection methods to daily life. Liu Wen [6] designed a human-friendly noise reminder device for quiet environments. Employing embedded development platforms and IoT cloud technology, they utilized a microcontroller on a development board to analyze collected sounds. Mechanically, corresponding controls were triggered, and noise reminders were facilitated through Android devices in collaboration with management.

Core to noise detection is classification based on standards, an essential concern in audio signal processing. In numerous fields such as speech recognition, music analysis, and environmental monitoring, sound signal classification and identification are necessary, driving the application of new technologies in noise detection. For instance, Wang Zongwei [4] applied robots for noise detection in non-factory environments. Gao Changfeng [5] studied the effect of self-supervised learning technology on robust speech recognition, improving feature representations insensitive to acoustic environment changes. Chen Peng [2] proposed a single-noise-source sound classification algorithm based on self-supervised learning principles, evaluating the effectiveness of the proposed algorithm through experiments.

3. Noise Detection Method for Factory Electromechanical Products

Current noise detection methods in manufacturing industries or for specific equipment involve complex and time-consuming design steps, requiring considerable hardware resources, as exemplified by Jiang Shanbin's [3] USB data collection, upper-lower computers, and circuit designs. Yet, machine learning-based noise detection research predominantly remains at the theoretical and laboratory testing levels, lacking industrial application cases. This study combines traditional electromechanical factory noise detection methods with machine learning, leveraging supervised learning advantages such as defined performance metrics, accuracy, precision, and recall. It simplifies hardware designs for noise collection and classification, employing a basic sound collector, Python programs for noise classification, reducing manual noise detection work. Leveraging extensive data collection for continuous model training improves detection accuracy. This low-cost, structurally simple system holds promise for widespread future implementation and usage.

4. Proposal for Deep Learning Integration

The integration of machine learning (ML) and deep learning (DL) techniques for online noise detection in electromechanical products is a valuable research topic. Key research steps involve hardware procurement and installation, data collection and preprocessing, feature extraction, machine learning classification, deep learning classification, model evaluation and comparison, and model optimization and improvement.

4.1. System Components

The initial step involves using a Logitech USB microphone to capture sound during product operation. This capacitor microphone, equipped with dual capacitive sound heads, efficiently captures sound from various angles, providing low-cost and high-quality recordings. The system diagram is depicted below (Figure 1), illustrating the sound data collection through the microphone for environmental perception. The mechanical part, primarily driven by PLC, controls the valve at the functional detection station in response to the analyzed sound data.

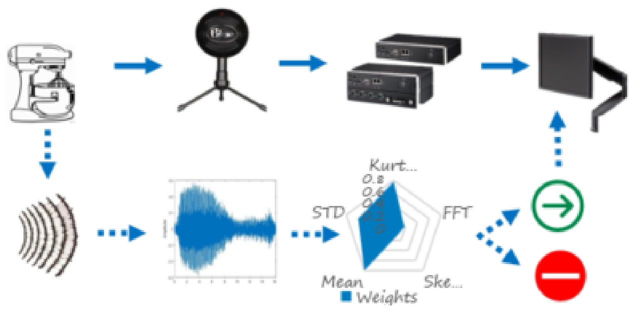

Figure 1. System Diagram

4.2. Data Collection and Preprocessing

In the laboratory, we meticulously labeled samples of both normal and abnormal sounds. Subsequently, we created 45 sets of metadata comprising 5 categories, defining data format, structure, quality, and annotation.

We utilized the WAV (Waveform Audio File Format) as our sound data format, widely used in Windows systems for lossless audio. WAV files digitally store raw audio data, ensuring high audio quality. This format supports various sampling rates and bit depths, enabling seamless audio editing and processing without compromising quality. During data loading, the torchaudio library was employed to load individual sample files, returning numpy arrays containing audio signals and sampling rates.

Noise signal preprocessing, also known as audio signal preprocessing, involves essential processing of audio signals before operations such as identification, classification, noise reduction, and enhancement. Common audio signal preprocessing methods include:

Noise Reduction: Removing background noise or other interference signals to enhance signal quality and reliability. Common noise reduction methods encompass filtering, spectral subtraction, and wavelet transformations.

Frequency Balancing: Adjusting the coloration and balance of audio signals by equalizing energy across different frequency components. Common frequency balancing methods include equalizers and filters.

Gain Control: Adjusting signal volume and intensity to ensure appropriate signal levels and stability. Common gain control methods include compression, limiting, and amplification.

Feature Extraction: Extracting essential features from audio signals, such as frequency, amplitude, time-domain characteristics, frequency-domain characteristics, for subsequent analysis and processing.

Data Normalization: Standardizing audio signals to a suitable range for ease of subsequent processing and analysis. Common data normalization methods include maximum value normalization and z-score normalization.

Different from conventional signal processing methods, we employed data augmentation techniques to enhance audio classification capabilities in data preprocessing. This technique plays a crucial role in subsequent feature extraction and model robustness, creating new data instances to increase training by

DOI: 10.54254/2755-2721/48/20241648

introducing noise disturbances to the original data, such as altering audio sampling rates, adding noise, etc., expanding the dataset size, and improving model generalization. We created the AudioUtil class to load audio data and perform data augmentation, enhancing further classification performance, primarily through 5 methods:

Resample: Standardizing all audio by converting them to the same sampling rate for uniform dimensions.

Rechannel: Converting mono files to stereo by duplicating the first channel into the second channel.

Padd or Truncate: Adjusting all audio samples to the same length by extending or truncating their duration.

Frequency Mask: Simulating a more realistic auditory environment on the spectrum graph through a combination of time and frequency masking, enhancing robustness and improving subsequent classification tasks.

Time Mask: Similar to frequency masking but employing random vertical bars on the time axis of the spectrum graph, enhancing diversity and robustness of speech signals to improve classification performance.

Time Shift: Shifts the original audio signal by a random amount to the left or right. The value at the end is wrapped around to the beginning of the transformed signal.

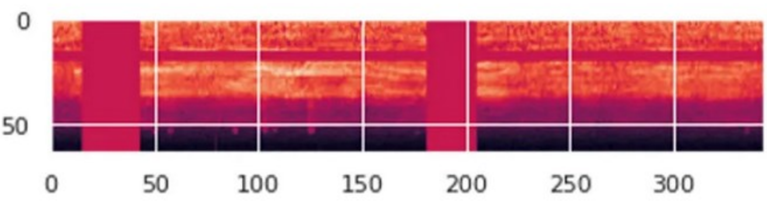

Figure 2. Example of Time and Frequency Masking Processing

4.3. Audio Feature Extraction

Audio signal feature extraction involves extracting relevant feature information from raw sound signals for subsequent analysis and processing. Common features of sound signals include several aspects:

Frequency Features: Include fundamental frequency, resonant peaks, spectrograms. The fundamental frequency is the base frequency of a sound signal, resonant peaks are strong spectral peaks, and spectrograms depict the distribution of sound signals in the frequency domain.

Time Domain Features: Encompass amplitude, energy, zero-crossing rate. Amplitude refers to the signal's magnitude, energy denotes the signal's energy within a specific time, and zero-crossing rate indicates the number of times a signal crosses zero.

Frequency Domain Features: Consist of spectral mean, spectral entropy, frequency moments. Spectral mean is the signal's average energy value in frequency, spectral entropy is the entropy value of the signal's spectral density function, and frequency moments are the signal's moment values at different frequencies.

Endpoint Detection Features: Used to identify the start and end positions within sound signals. Common features include short-time zero-crossing rates, short-time energy, etc.

Non-linear Features: Comprise entropy, spectral reduction, fractal dimensions. Entropy measures signal complexity, spectral reduction reduces signal dimensions in frequency, and fractal dimensions describe the signal's fractal characteristics.

Audio feature extraction is a crucial step in audio signal processing. Common methods include Short Time Fourier Transform (STFT), Discrete Cosine Transform (DCT), Mel Spectrogram, etc. These basic features have evolved into various features tailored for different tasks.

1. Pre-processing: Includes pre-emphasis and framing. Pre-emphasis balances the spectrum, emphasizing high frequencies, while framing divides audio signals into short-time frames, usually 20-30ms each with some overlap between frames.

DOI: 10.54254/2755-2721/48/20241648

2. Window Function: Applies a window function (like the Hamming window) to each frame, reducing spectral leakage at frame edges.

3. Fast Fourier Transform (FFT): Performs FFT on each frame, converting signals from the time domain to the frequency domain.

4. Mel Filter Bank: Passes the spectrum through a Mel Filter Bank to mimic human ear frequency perception. This bank typically contains 20-40 triangular filters evenly distributed on the Mel scale.

5. Logarithmic Energy: Computes the logarithmic energy of each filter output.

6. Discrete Cosine Transform (DCT): Applies DCT to the logarithmic energy to extract cepstral coefficients. Typically retains the first 12-13 coefficients containing the primary signal information.

We employ Mel Spectrograms to capture fundamental features of audio. It serves as the most suitable method for inputting audio data into deep learning models, particularly in speech recognition. It simulates human auditory perception of frequencies by converting the spectrum into Mel scales. Its advantages include:

Simulating Human Auditory Properties: Mel frequency spectrum maps frequencies into a range associated with human hearing perception. The Mel scale aligns more closely with how the human ear perceives sounds, making it highly suitable for processing sounds from electro-mechanical products.

Low Dimensionality of Features: Mel frequency spectra transform audio signals into feature vectors with lower dimensions, typically ranging between 10 to 20, compared to other frequency analysis methods. This leads to faster computation of Mel frequency spectra and easier handling and analysis of feature vectors.

The formula for converting ordinary frequencies into Mel frequencies is expressed as follows:

$$Mel(f) = 2595 * \log10(1 + f / 700) \quad (2\text{-}1)$$

Robustness Against Noise: Mel frequency spectra utilize a logarithmic energy spectrum during feature computation, balancing the energy contributions across different frequency bands. This robustness enhances the capability of feature extraction in noisy environments, making Mel frequency spectra excel in extracting features from noisy settings.

Therefore, Mel Spectrograms can capture the basic characteristics of speech signals and exhibit good robustness across various speech signals.

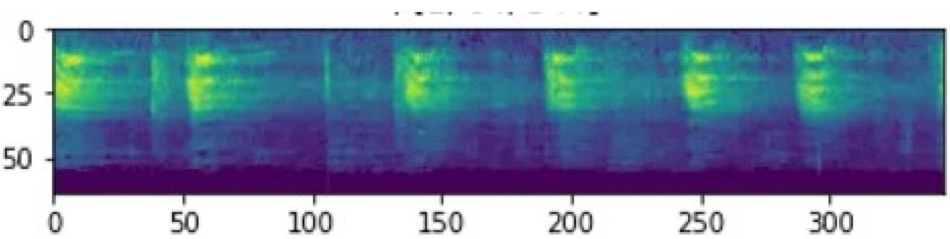

Figure 3. Example of Mel Spectrogram for Data Samples

4.4. Built-in Data Loader

The built-in DataLoader in PyTorch is a class used to batch-load data from custom datasets. It offers a simple way to batch-fetch data, shuffle, and perform batch processing. Typically, it requires specifying parameters like batch size and the number of workers. Using the built-in DataLoader facilitates handling large-scale datasets and enables operations such as GPU-accelerated training.

4.5. Establishment of Neural Network Model

A Convolutional Neural Network (CNN) classification architecture based on the PyTorch framework is employed for audio signal classification. This model comprises four convolutional blocks, each generating feature maps. The input audio signal is in batch format, with a specific shape (batch_size, num_channels, Mel freq_bands, time_steps). Each CNN layer applies its filters to increase the image's depth. With the application of kernels and strides, the width and height of the image decrease. Subsequently, pooling and flattening operations are performed, and the result is fed into a linear layer. The linear layer outputs prediction scores for each category.

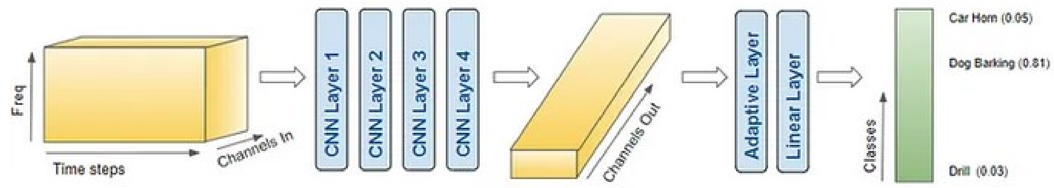

Figure 4. Example of the Neural Network Model

4.6. Experimental Design and Results

Through supervised learning focused on factory testing, we prioritize accuracy, efficiency, consistency in judgments, and consider computational operability as key evaluation metrics for the model.

Table 1. result comparing

	Accuracy(N=500)	Time*(N=500)*	Loss
Manual	80%	4 hrs	0.2
AI Model	99%	16 mins	0.01

5. Discussion of Results

Multiple rounds of model parameter adjustments and training have significantly improved noise detection results to 99%, a 19% increase compared to manual judgments. Given the lower investment cost of deep learning methods, compared to existing manual inspection methods, it enhances detection speed, accuracy, and reduces costs.

While we have implemented a series of product noise detections on one production line, limitations persist. For extending this method to other products, supervised learning requires manual data annotation and model adjustments. In 2024, our project will continue advancing. Currently, deep learning technology is only applied to noise identification and classification. Future efforts will delve into the correlation with component failures, further noise source AI detection and analysis, fault diagnosis, optimizing product design, reducing defect rates, and transitioning data to cloud-based systems for remote anomaly alerts.

References

[1] Wang, F., "Defective Bearing Noise Detection Research," 2015, North China University.
[2] Chen, P., "Research on Self-supervised Noise Classification Algorithm."
[3] Jiang, S., & Zhu, S., "Design and Application of USB Data Acquisition Card in Bearing Detection," Journal of Jiangnan University (Natural Science Edition), 2006, 5(5), pp. 531-535.
[4] Wang, Z., & Zhang, D., "Research on Noise Detection and Analysis of Substation Inspection Robots," Electronic Testing, 2018(23), pp. 51-52+55.
[5] Gao, C., Cheng, G., & Zhang, P., "Consistency Self-supervised Learning Method for Robust Automatic Speech Recognition," Journal of Acoustics, 2023, 48(3), pp. 578-587.
[6] Liu, W., et al., "Design of Humanized Noise Detection Prompt System," IoT Technology, 2019, 9(1), pp. 18-20.
[7] Yu, C., "Research on Vibration and Noise Control Technology of Metallurgical Machinery," Metallurgy and Materials, 2023, 43(09), pp. 83-85.