Master of Science in Mechanical Engineering

A Sound-based Real-Time Machine Monitoring System in Edge Computing

음향분석을 통한 에지 컴퓨팅 기반 실시간 제조장비 모니터링 시스템

February 2023

Graduate School of Mechanical Engineering
Seoul National University

Hyucksoon IM

A Sound-based Real-Time Machine Monitoring System in Edge Computing

Sung-Hoon Ahn

Submitting a master's thesis of
Mechanical Engineering

October 2022

Graduate School of Mechanical Engineering
Seoul National University

Hyucksoon IM

Confirming the master's thesis written by
Hyucksoon IM

December 2022

Chair _____(Signature)
Vice Chair _____(Signature)
Examiner _____(Signature)

Abstract

As the importance of data utilization increases, manufacturing industries aim to become smart factories through meaningful information from various sensors. The priority is to collect data generated in fields and monitor the machine. Among many types of data, sound is not only easily acquired but also economic data containing much site information. However, there are many restrictions to be addressed in the central data center because sound sources are relatively large and complex to protect privacy. We propose a sound-based machine monitoring system embedded in an edge computer to overcome these limitations. The proposed system consists of Jetson Nano, an edge computer, and a microphone array Respeaker v2.0 for data acquisition. The recorded sounds are augmented with random mixing and amplitude adjustment then optimal parameters are selected. To reduce the computational cost, a model is designed to be small but capable of fast inference and high accuracy. Considering the noise generated in the environment, the model is trained using a dataset generated by an autoencoder network. This system is developed and verified in a lab environment and then demonstrated at a manufacturing site. As a result, this model monitors the operation state of the target machine with an accuracy of 93% and an inference time of 1.1 seconds in a noisy environment.

Keyword : sound, real time, edge computer, IoT, machine monitoring

Student Number : 2021-21595

Table of Contents

Chapter 1. Introduction

1.1. Background

The fourth industrial revolution is leading the breakthrough growth of manufacturing industries around the world by analyzing information generated from sites through advanced technologies such as the internet of things (IoT), Artificial Intelligence (AI), and blockchain [1−4]. Since these changes affect enterprises' growth and social problems such as global warming and environmental pollution, many studies are being conducted in industries and academia [5]. However, while large enterprises can respond to this flow, converting small and medium−sized enterprises (SMEs) face a lack of technology and data and a financial burden [6, 7]. With the concern of SMEs, Jung *et al.* proposed an appropriate smart factory to build a system that performs proper functions, is available for purchase, and is easy to apply on−site [8].

A monitoring system is essential for establishing a smart factory [9]. Monitoring of processing equipment increases productivity and is based on managing and predicting the health of machines [10−12]. However, state−of−the−art machines are equipped with monitoring and analyzing systems. Legacy equipment has fewer features like these. For this reason, many studies have proposed to remotely monitor the state of old equipment by combining IoT, communication, and AI technologies with various sensors, including vibration, sound, electricity, and vision [13]. Table 1.1 compares clustered sensors by

detection method. Kim *et al.* [14] monitored a 3-axis computer numerical control (CNC) milling machine through a low-cost camera and open-source technologies. This system led an SME to reduce operation time and energy consumption. Jung *et al.* [15] implemented an IoT-based power monitoring system for sewing machines and demonstrated it in a garment manufacturing factory. Consequently, the power monitoring system improved productivity as well as work efficiency. Kim *et al.* [16] proposed a sound-based machine monitoring system and demonstrated it in a small factory.

Table 1.1. Comparison of sensors used in an industrial site. Radio detection and ranging (RADAR) (***: Good / **: Normal / *: Bad)

Index	Motion [17]	Vibration [18]	Optical [19]	Energy [20]	Acoustic /sound [16]
Sensor type	RADAR	Accelero-meter	Camera	Power-meter	Micro-phone
Sensor cost	*	**	*	**	***
Simple to install	***	*	***	*	***
Detection range	***	*	***	*	***
Data size	*	**	*	***	**
Privacy security	**	***	*	***	*
Noise immunity	**	*	***	**	*
Process complexity	*	**	*	**	**

However, IoT-based systems face several challenges when data moves from the sensing location to a centralized server: latency, scalability, and privacy [21]. For example, a vision-based application in an autonomous vehicle system may need three-dimensional images as raw data. However, the data size is so large that it is impossible to proceed with the inference process after going through the central server in real-time [22]. Moreover, the more IoT devices connecting to the central server, the more traffic will increase, and eventually, a bottleneck occurs, making data transmission on the network inefficient. Lastly, transferring private data such as face photos and recorded voices can cause an infringement of personal information [23].

Edge computing is considered a suitable solution for addressing these obstacles by decentralizing the workloads of the main server. To address the latency challenge, edge devices collect and compute data to reduce end-to-end latency and thus enable real-time services. In terms of scalability, the number of edge computers increases in proportion to the number of users; the bottleneck is reduced. Lastly, edge computer processes raw data locally; thus, privacy and security attacks are prevented. Although the advantages of edge computing, edge computers have relatively low computer power [24], so it is limited to embed a deep learning model. Therefore, it is required to develop an edge computing system equipped with an optimal deep learning model suitable for the specific purpose.

1.2. Purpose of Research

In a factory environment, machine sounds contain meaningful information such as operation rate, machine state, prognostics, and health management (PHM). In this study, we propose a sound-based machine state monitoring system and embed this system in an edge computer described in Figure 1.1. We aim to develop a small but accurate classification model so that it can operate on an edge computer. Then, we demonstrate the system in different manufacturing environments: a laboratory environment for prototyping and a manufacturing industry for small-quantity batch production.

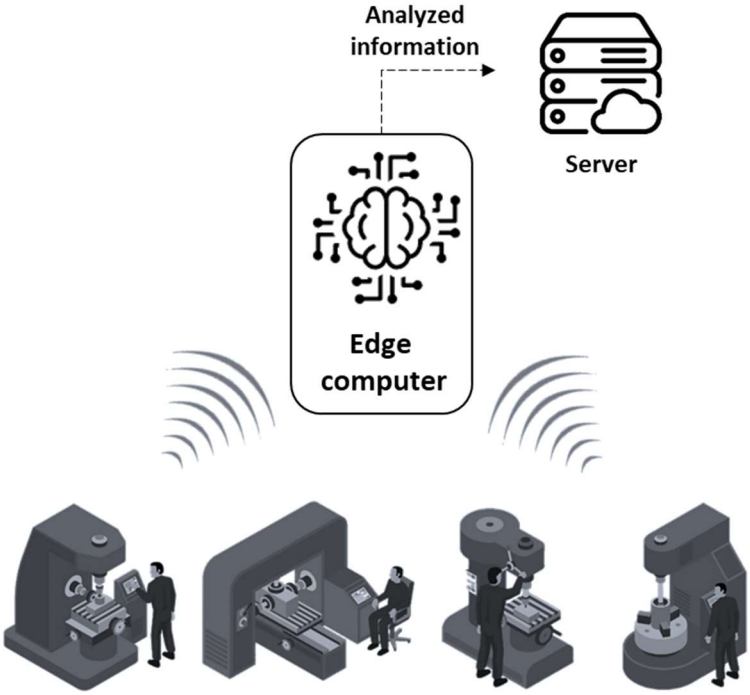


Figure 1.1. Schema of the proposed sound-based real-time monitoring system.

Chapter 2. Sound analysis and processing

2.1. Physical property of sound

The sound waves provide us with multifactorial information such as frequency, intensity, and timbre with identified characteristics. In physical, sound is a type of energy produced by the vibration of an object through a transmission medium such as a gas, liquid, or solid. When vibration occurs, air molecules oscillate through space, tending to bump into each other. This movement generates waves consisting of compression parts with relatively more molecules and rarefaction parts with relatively fewer molecules.

A waveform consists of frequency, amplitude, and phase. Frequency refers to the number of oscillations through the medium per unit of time. Amplitude indicates the magnitude of energy with respect to atmospheric pressure. When molecules in a medium are significantly vibrated, it has a high magnitude. Lastly, phase specifies the position of the waveform.

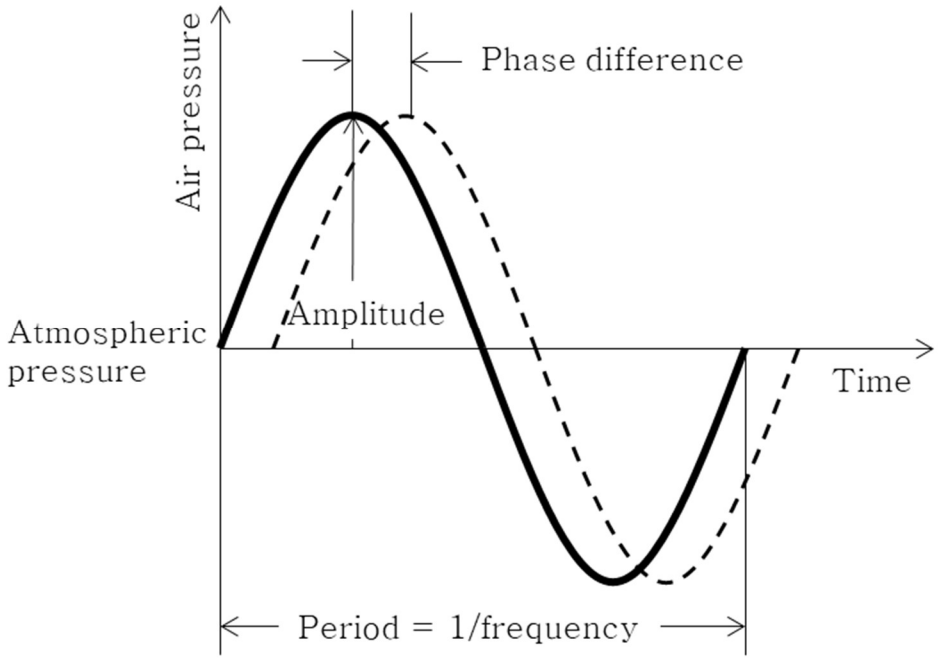


Figure 2.1. The sound wave consists of amplitude, frequency, and phase. In particular, the amplitude represents the degree to which the molecules in the air are concentrated.

2.2. Fourier transform

Complex sound waves represented by the time domain can be decomposed into their frequency components by mathematical transformations called Fourier transform. This method is based on a principle; every complex sound is represented as a sum of sinusoid waves with distinct frequency, amplitude, and phase. In digital signal processing, since N data are sampled at finite intervals, the complex number X_k of a frequency k is expressed as follows:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi n\frac{k}{N}} \qquad \cdots \text{ eq 2.1}$$

From the formula above, although the Fourier transform has the advantage of being analytical through three components, it is impossible to represent a continuous perspective because time information disappears. A Short Time Fourier Transform (STFT) is presented to complement the Fourier transform by multiplying the entire signal by a window function ω and dividing it into short segments of the time series. Windows have the same length of the number of data N, and it is called frame length. So, at the current frame number m with the hop size H, the value of the frequency k in the current window m is expressed as:

$$S_{(m,k)} = \sum_{n=0}^{N-1} x(n+mH) \cdot \omega(n) \cdot e^{-i2\pi\frac{k}{N}} \qquad \cdots \text{ eq 2.2}$$

However, the measured signal is leaked at the end of the window because the processed signal is not an integer number of periods, so the spectrum is smeared. To minimize the leakage effect, the weights of samples at both ends of a frame should be eliminated by convoluting a shaped window with an amplitude varying smoothly toward zero at the endpoints of the signal. Then, the window is shifted slightly, overlapping the previous window to compensate for reduced weights at the edges, and its overlapping length is called hop length. This pipeline converts the digital signals in the time domain into the frequency domain from which frequency features are extracted. Finally, the sound analysis is performed by restructuring the spectrums computed in frame units into a vector or matrix form.

2.3. Spectrogram and filter bank

Spectrogram refers to a 3D image representing the spectrums of frequencies over time, which is suitable for analyzing sound data through machine learning. In a spectrogram, each column represents the frequency and magnitude that STFT calculates in each window.

Figure 2.2 shows three types of digital filter banks that can be applied to each frequency's weights to improve the analysis efficiency by frequency selectivity. A Mel filter bank, a triangular filter bank, provides higher frequency resolution at low-frequency range, while lower frequency resolution at high-frequency range in logarithmic scale, imitating the human ears perception of sound.

$$m = 2595 \cdot \log\left(1 + \frac{f}{500}\right) \qquad \cdots \text{ eq } 2.3$$

$$f = 700(10^{m/2595} - 1) \qquad \cdots \text{ eq } 2.4$$

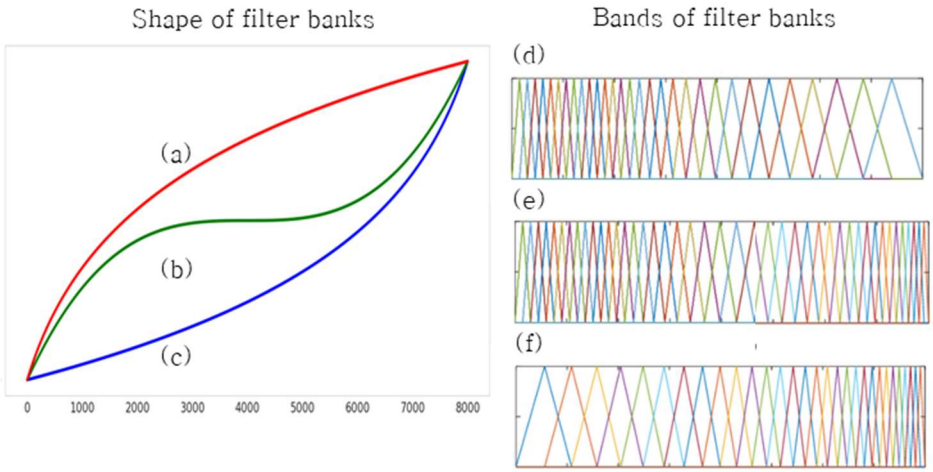


Figure 2.2. The shape of digital filter banks. (a) and (d) represent the log−mel scale, (b) and (e) represent the tangent scale, and (c) and (f) represent the exponential scale.

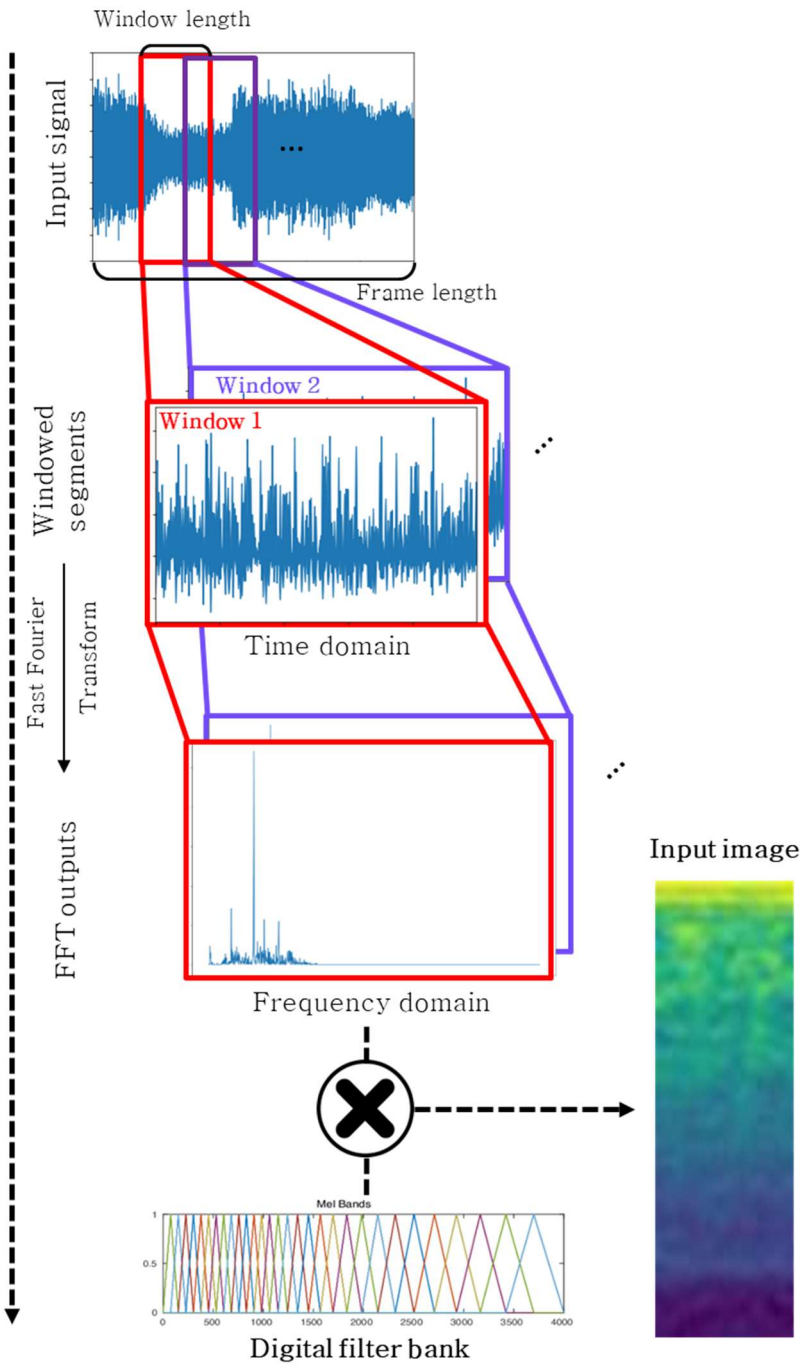


Figure 2.3. Process of Fast Fourier Transform for discrete signal.

2.4. Convolutional neural network

In the field of Deep Learning (DL), a Convolutional Neural Network (CNN) has shown significant research results in image analysis [25, 26]. The CNN was first presented by Yann *et al.* to overcome the limits of traditional models of pattern recognition and machine learning algorithms [27]. This promising network has been developed from 1989 until today in various fields, including speech processing [28], computer vision systems [29].

A CNN consists of convolution and pooling processes. In the convolution process, a kernel called a filter plays the role of the feature extractor. The kernel made of a grid of weights is computed as a dot product to the image with several setting parameters; grid size, stride, and depth. Grid size is the number of pixels for height and width. Each grid size of a kernel is generally odd for preventing image distortion by clarifying the position of the center pixel. A stride is the step size used for sliding the kernel on the image. The color of the original image data decides the depth. The depth is three for RGB images, while the depth is one for gray images. Next, we obtain a non-linear feature map by applying an activation function to the weights following:

$$h^k = f(W^k \cdot x + b^k) \qquad \cdots \text{ eq 2.5}$$

Whereas a pooling process is performed after convolution to prevent overfitting during training by removing the feature map's location dependency and reducing the computational cost through down-sampling. Max or average pooling methods are generally used. While max pooling extracts only the largest value from the input image by a patch with a specific grid size and stride, average pooling extracts the average value of the values in the patch.

After a bunch of convolution and pooling processes, the last 2D image from which high-level features are extracted is flattened as a vector and then classified by an activation function such as softmax, sigmoid, or Rectified Linear Unit (ReLU). Finally, the prediction output for each class is deduced then the weights are optimized gradually based on error backpropagation with a proper loss function.

In this study, we employed a CNN-based architecture for data analysis, considering sound as a spectrogram image. This visualized sound represents the time, frequency, and amplitude of the x, y, and pixel value, respectively.

Chapter 3. Real-time sound monitoring system

3.1. Hardware of monitoring system

This chapter describes the hardware configuration used to monitor the sound of the designated machine in real-time in a noisy indoor space and a process for efficient data gathering, pre-processing, learning, and classification. As an edge computer, a Jetson nano (NVIDIA, USA) was chosen to serve as a classifier embedded with trained models and sound data recording, as described in Table 3.1. This edge device delivers 472 GFLOPs (Floating point Operations Per Second) of computing performance and a GPU mounted with only 5 watts of low power consumption. With these advantages, many domains prefer to utilize jetson nano, where AI or machine learning techniques are applied with edge computing. Specifications for the microphone array (Seeed studio, China) selected for sound data acquisition are shown in Table 2.2.

3.2. Data acquisition and preprocessing

We recorded machine sounds in a laboratory environment and manufacturing factory with noise, reflection, and diffraction for the generalization of the acquiring dataset as well as preprocessing parameters.

Table 3.1. Specification of used edge computer

Specification	Description
Name (company)	Jetson Nano (NVIDIA corp)
GPU	128-core Maxwell
CPU	Quad-core ARM A57
Memory	4 GB 64-bit
Dimension	69 x 45 mm^2
Appearance of Jetson nano 	

Table 2.2. Specification of mic array

Specification	Description
Name (company)	Respeaker Mic Array v2.0 (seeed studio)
Signal processor	XVF-3000 from XMOS
Microphone type	ST MP34DT01TR-M (Digital MEMS)
Sensitivity	- 26 dBFS
Signal to noise ratio	61 dB
Dimension	70 mm (Diameter)
Max sample rate	16 kHz
Appearance of respeaker mic array v2.0 	

The machining sound amplitude of each target machine was measured in decibels(dB), and a mic array was installed at a position where the sound volume was as similar as possible. Furthermore, the mic array is set parallel to the ground so that all mic arrays can record the sound of all equipment. The data acquired in this way are divided into clean data recorded by processing equipment in the absence of noise and various noises that may occur in the environment. Figure 3.1 shows the data augmentation process using parameters appropriate for the acquired raw data to prevent overfitting and enable real-time processing. This technique not only increases the amount of data but also considers variables of the physical properties of sound.

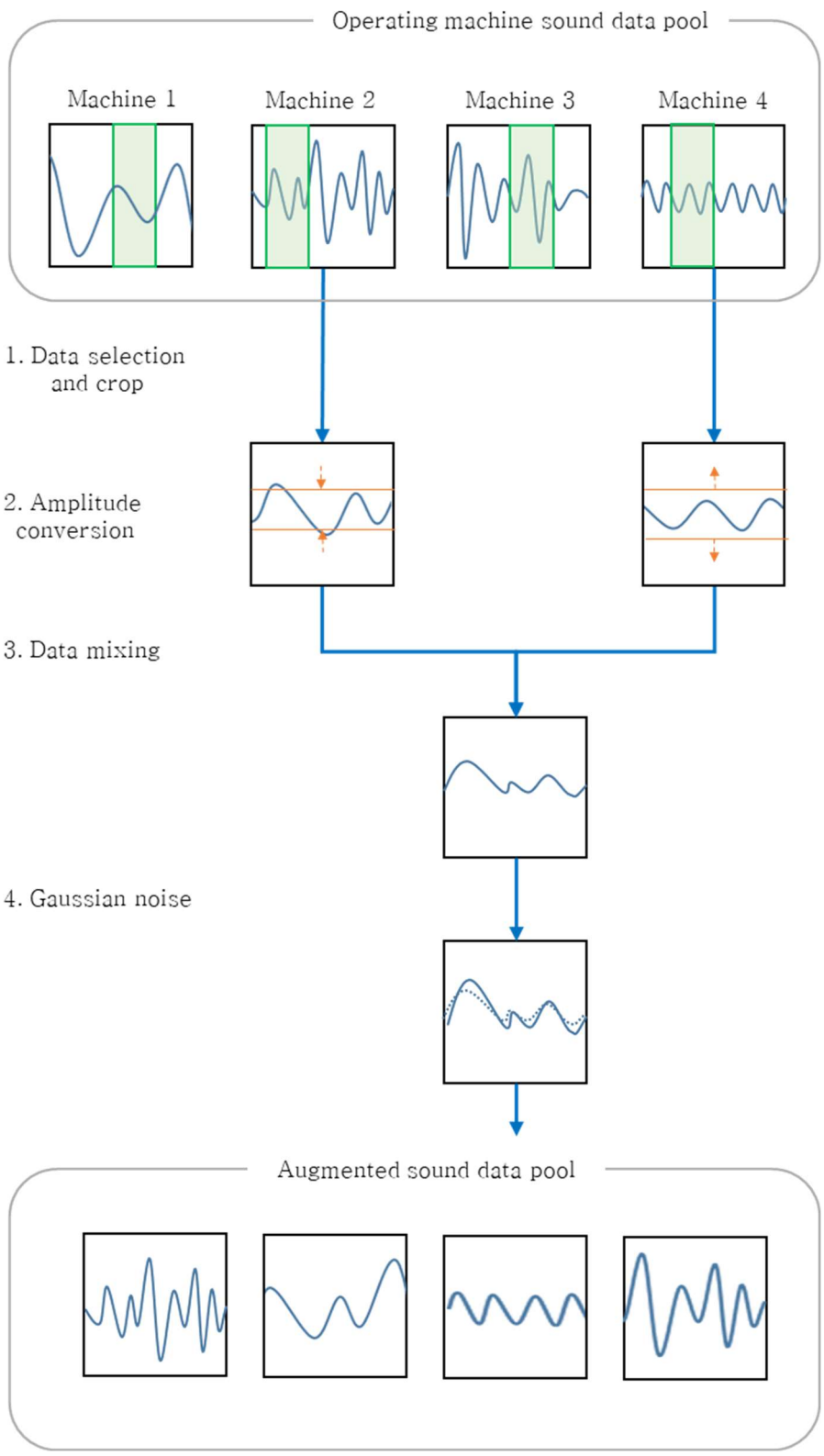


Figure 3.1. Data augmentation process.

At first, we crop an arbitrary time interval from the waveform of any amount of equipment. The selected waveforms are randomly multiplied by 0.1 to 1.1 times amplitude to prevent performance degradation due to a change in position between the mic array and the machines. Then the data are mixed and multiplied by gaussian noise. Finally, the augmented sound data pool is prepared. Next, STFT is performed using parameters including frame length, window length, and hop length, shown in Table3.3. We use the Hann window to minimize frequency resolution and amplitude deformation errors. A digital filter bank is selected to generate a spectrogram. Generally, a Log−Mel filter bank is used to emulate human ears, but in this study, we compare the performance of the log scale, exponential scale, and tangent scale. The spectrogram is along a vertical image with a 256x32x1 grayscale because frequency information is more characterized than time information. Figure 3.2 is the generated data through the above preprocessing and is fed as input data for inference and model training.

Table 3.3. Parameters to compare the performance of models.

Parameter	Values used in this study
Frame length (seconds) (Number of data)	0.2, 0.5, 0.8 (3,200, 8,000, 12,800)
Window length ratio to frame length	0.1, 0.3 0.5
Filter bank type	Log−mel, Exponential, Tangent
Sampling rate, image size	16 kHz, (256, 32)

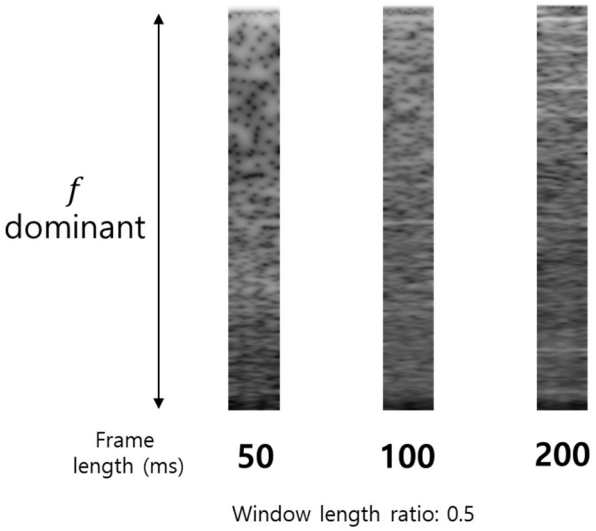


Figure 3.2. Generated images(256x32x1) with different frame lengths of 50, 100, and 200 milliseconds and a window length ratio of 0.5.

3.3. Design of a classification model

To infer in real time using edge computers, a model that accurately deduces while reducing the number of parameters and FLOPs used for updates is needed. We adopt a Depthwise Separable Convolutions(DSC) technique that combines depthwise and pointwise convolution from the mobilenet presented by Howard *et al.* [30], which is the most widely used CNN model for analyzing images. Figure 3.3 shows the structure of the proposed network using DSC.

Depthwise convolution train filters use only spatial information for each channel by proceeding with convolution in the spatial direction except for the channel direction. Simultaneously, we compress the channel using pointwise convolution to perform convolution in the channel direction rather than the spatial direction.

Let the size of the squared kernel of the input be D_K^2 and the size of output feature maps by D_G^2, and each has the number of channels M and N, respectively. The ratio of the number of parameters and Flops of the standard CNN and the DSC is shown in eq 3.1 and eq 3.2.

$$\frac{P_2}{P_1} = \frac{(N + D_K^2) \times M}{N \times D_K^2 \times M} \qquad \cdots \text{ eq 3.1}$$

$$\frac{F_2}{F_1} = \frac{D_G^2(D_K^2 + N) \times M}{N \times D_G^2 \times D_K^2 \times M} \qquad \cdots \text{ eq 3.2}$$

where, P_1 and F_1 are the number of parameters and the floating points of standard CNN, respectively and P_2 and F_2 are the number of parameters and floating points of DSC.

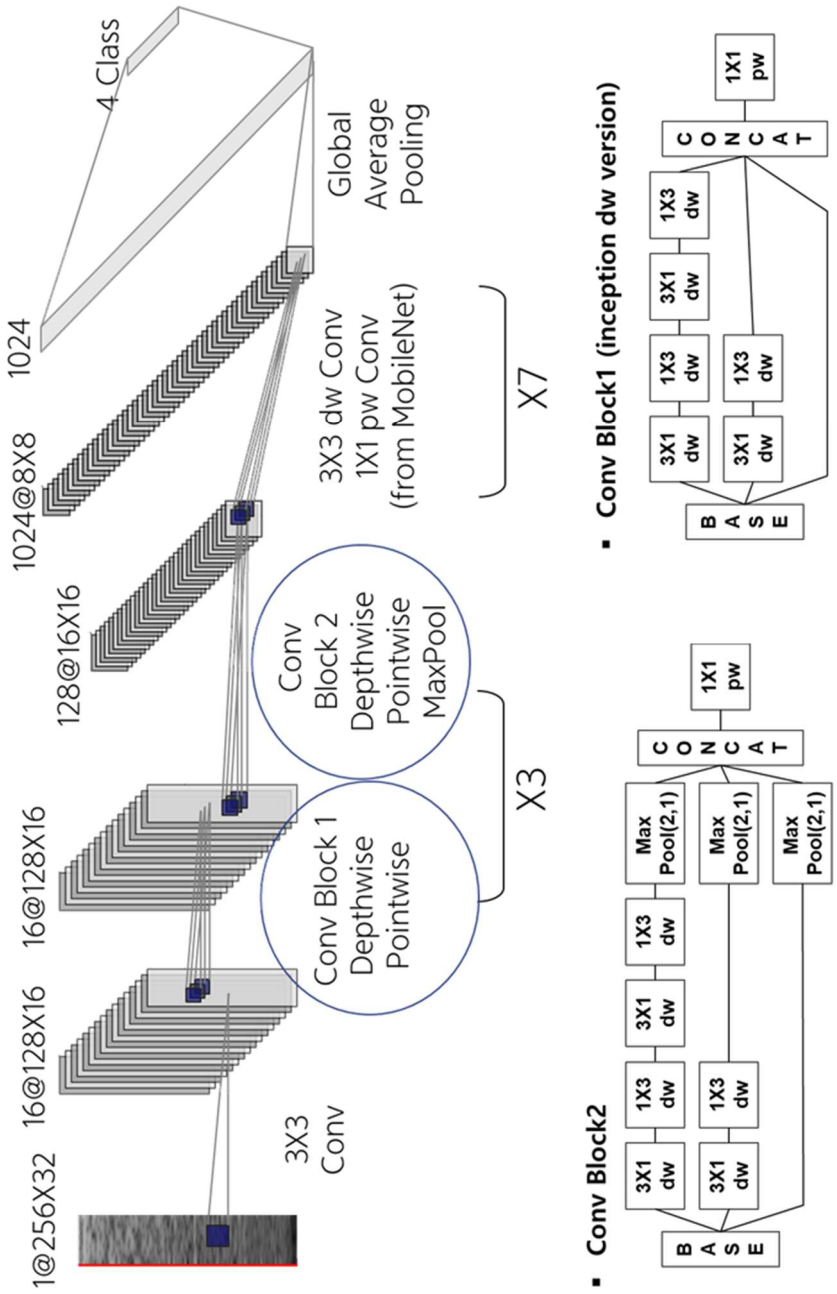


Figure 3.3. Structure of the proposed network. Mobilenet is referred to as a backbone network.

3.4. Design of autoencoder network

Sound analysis is vulnerable in the manufacturing areas where unexpected noise, such as working fans, human voice, and vehicle sound, occurs. To prevent performance degradation to noise, we adopted the autoencoder technique.

Autoencoder [31] is a network of selecting and extracting features of input data to learn data encoding in an unsupervised manner. There are three parts in an autoencoder: encoder, decoder, and bottleneck. Firstly, the encoder compresses the input data into an encoded representation while decreasing the dimension. At the end of the encoder, the bottleneck performs to contain the most crucial feature of the network. Lastly, a decoder consisting of the same architecture as the encoder reconstructs the compressed data and generates data from its latent attributes. The network parameter of the autoencoder is trained by comparing the output image with ground truth in a supervised manner.

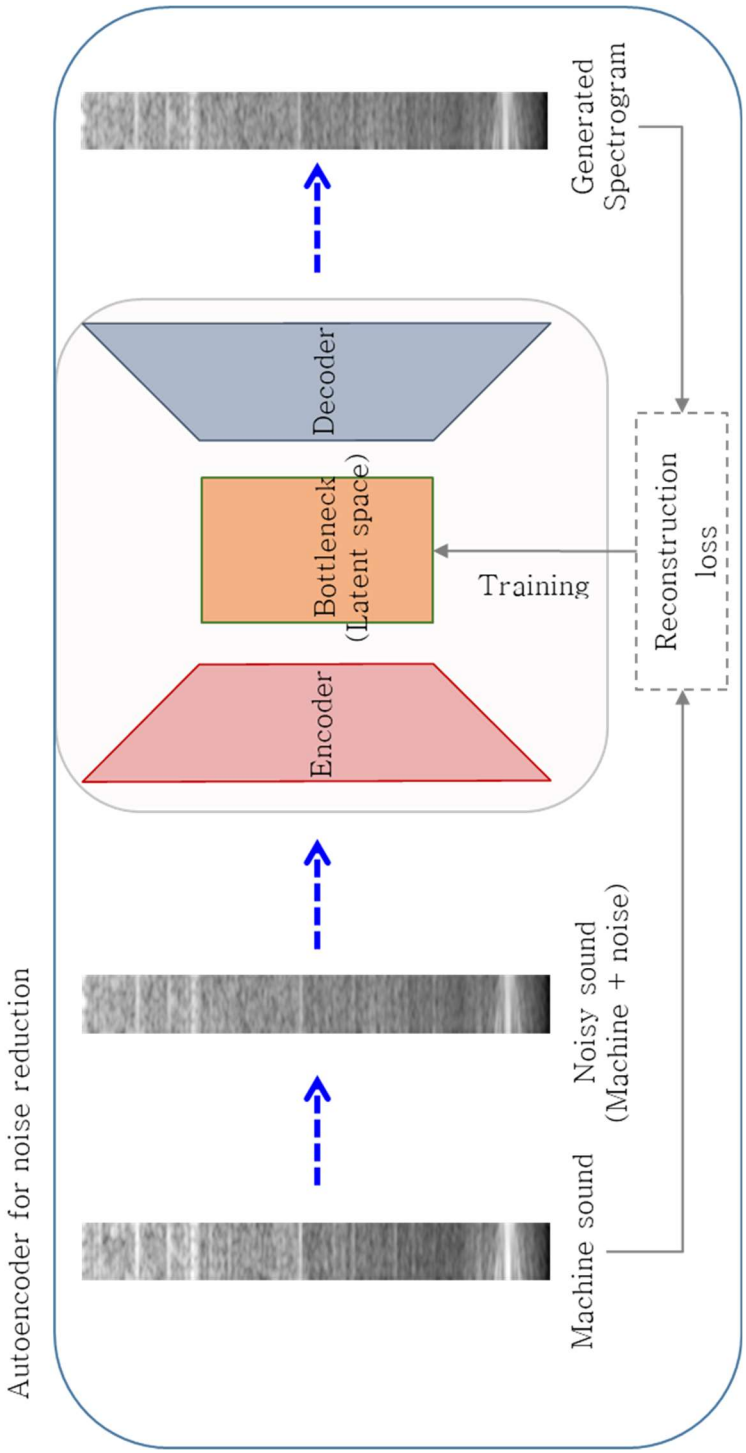


Figure 3.4. Process of training for autoencoder network and proposed network.

A dataset consists of a pure sound of equipment acquired in a noise-controlled environment and a dataset of noise in the site, with the preprocessing parameters presented in Table . We trained the autoencoder model with a mean squared error as a loss function. Figure 3.5 shows samples of spectrograms of purified machine sound, a machine with noise sound, and generated sound by autoencoder in order.

Table 3.4. Processing parameters for training the autoencoder.

Parameter	Value
Frame length (Seconds) (Number of data)	0.5 (8,000)
Window length ratio to frame length	0.3
Filter bank type	Log−mel
Sampling rate	16 kHz
Number of generated data	20,000
Image size	256, 32

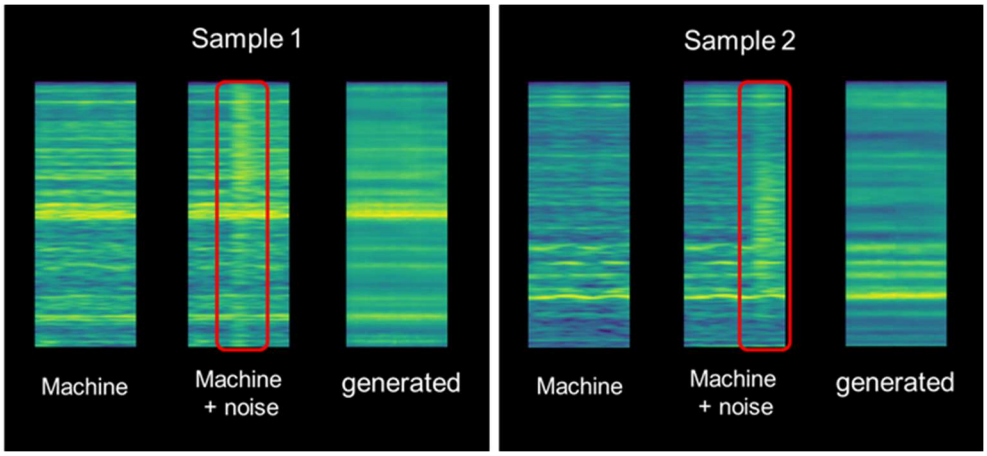


Figure 3.5. Two samples of images of machine sound, the machine with random noise, and generated.

Chapter 4. Results

To verify the performance of the developed sound monitoring system, we conducted tests in two places, including laboratories and the manufacturing industry, with different target machines, noise, and areas.

4.1. Case 1 – Lab environment for manufacturing prototyping

Figure 4.1 shows the laboratory where researchers conduct experiments and prototypes with small machines such as drills, mini lathes, 3D printers, collaborative robots, and laser cutters. Among these machines, we decided to monitor the sound of a mini lathe, band saw, CNC milling, and drill press which are conventional pieces of equipment.

We evaluated the performance of this system with data set according to the scheme in Figure 4.1 and Table. At first, we compared the performance of each combination of the preprocessing parameters. A performance comparison of each parameter is shown in Figure 4.2. When the frame length was 0.5 seconds and 0.8 seconds, similar results were obtained with an accuracy of about 0.94%, but the longer the frame length, the longer the inference time, so 0.5 seconds was selected. In addition, the window length ratio of 0.3 seconds was selected as the optimal parameter.

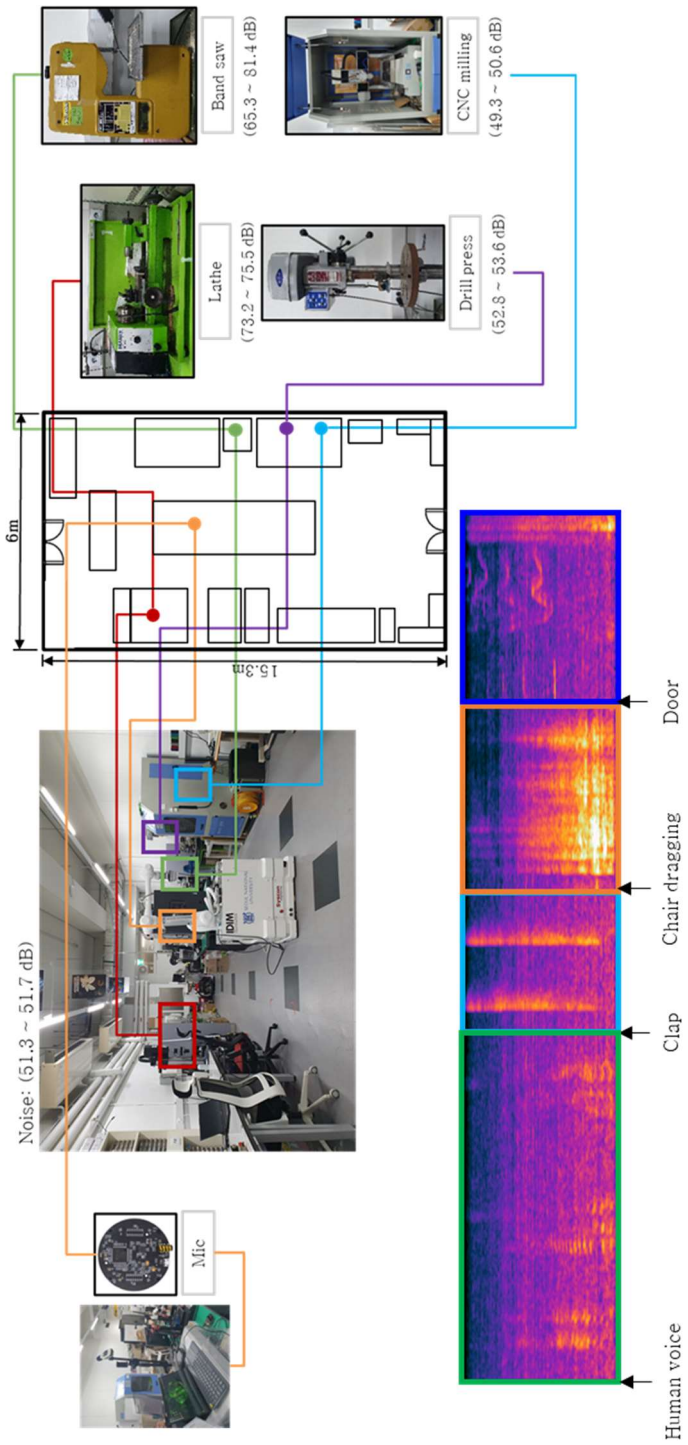


Figure 4.1. Layout of a real-time sound monitoring system in the laboratory and a spectrogram of noise

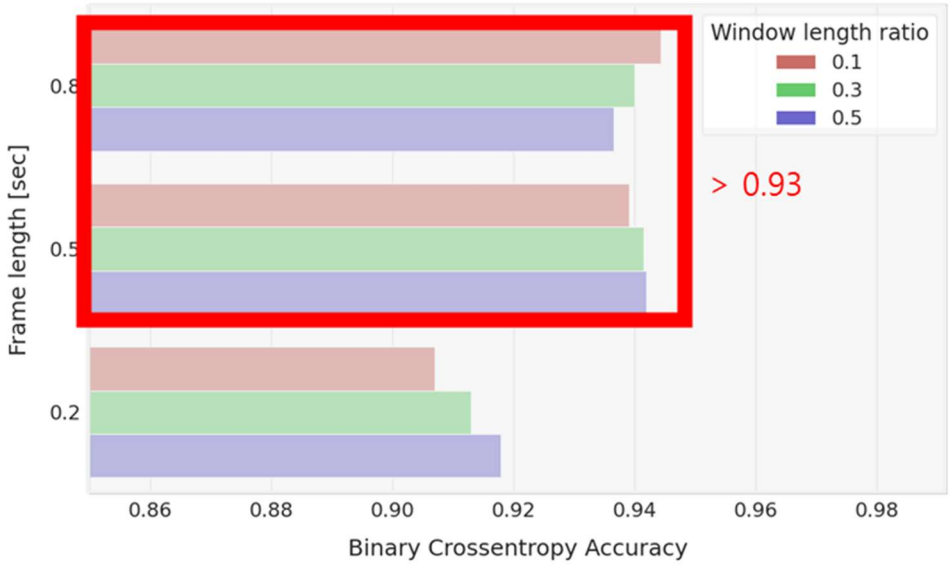


Figure 4.2. Comparison of model performances according to different frame lengths and window length ratios.

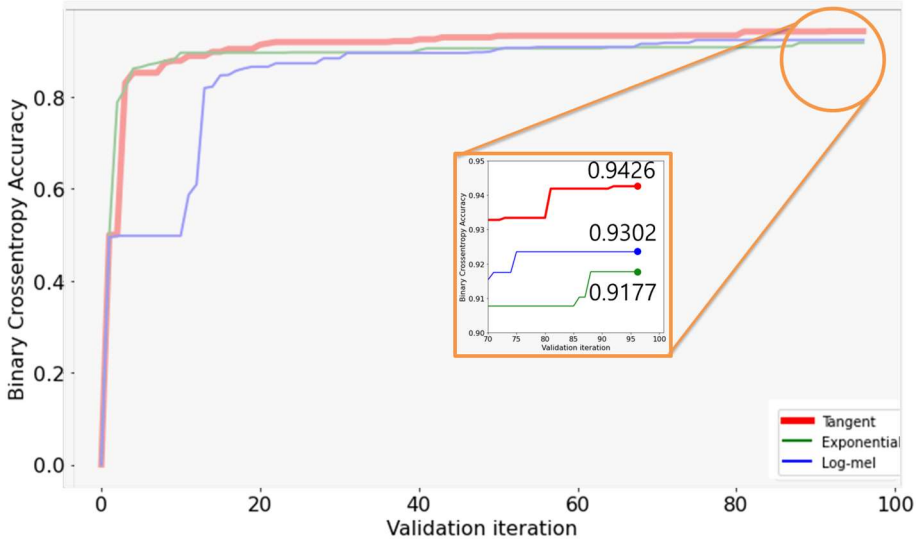


Figure 4.3. Comparison of binary cross entropy accuracy of each filter bank.

We compared the performance of the three filter bank types using the optimized parameter selected. The log-Mel filter bank increases the resolution for the relatively low-frequency range, while the exponential filter bank increases the resolution for the high-frequency range. Moreover, instead of decreasing the resolution for the intermediate frequency range, the tangent filter bank focuses on low-frequency and high-frequency ranges. Figure 2.2 shows that the accuracy differs according to each filter bank type, proving that weight selection for each frequency range is essential.

With the proposed network, we trained a model using 20,000 generated data, 100 ms frame lengths, and a 0.25 window length ratio. Then, we compared the training results, including binary accuracy, precision, recall, number of parameters, and FLOPs, with the baseline models. Table proves that the number of parameters and FLOPs of the proposed network was lower than the existing networks, which means that a fast and accurate model with an inference time of 0.3 seconds and being accurate with low memory usage.

The proposed model classified the machine sound with a 0.95% accuracy. However, in the noisy environment, its accuracy was significantly reduced to 0.81~0.87%. To consider the noise effect, we trained the proposed classification model with the dataset generated by autoencoder. Figure 4.4 compares the accuracy of models with the model trained by only purifying data in a noisy environment. With slight noise, including sneezing, clapping, door, and char dragging, the autoencoder model, improves the accuracy from 0.87% to 0.95%. On the other hand, in a loud, noisy environment

where non-target machines operate, the accuracy increased from 0.81% to 0.95%. The total processing time is about 1.1 seconds, of which 0.5 seconds is for data sampling, 0.3 seconds is for passing the autoencoder network, and 0.3 seconds is for inference time.

Table 4.1. Performance comparison of proposed network with reference networks.

Model / Performance	VGG11	MobileNet	Proposed Network
Binary accuracy	0.978	0.978	0.986
Precision	0.493	0.493	0.495
Recall	0.984	0.984	0.987
Number of parameters	24.4 M	3.2 M	2.6 M
FLOPs	82.4 M	11.6 M	5.2 M

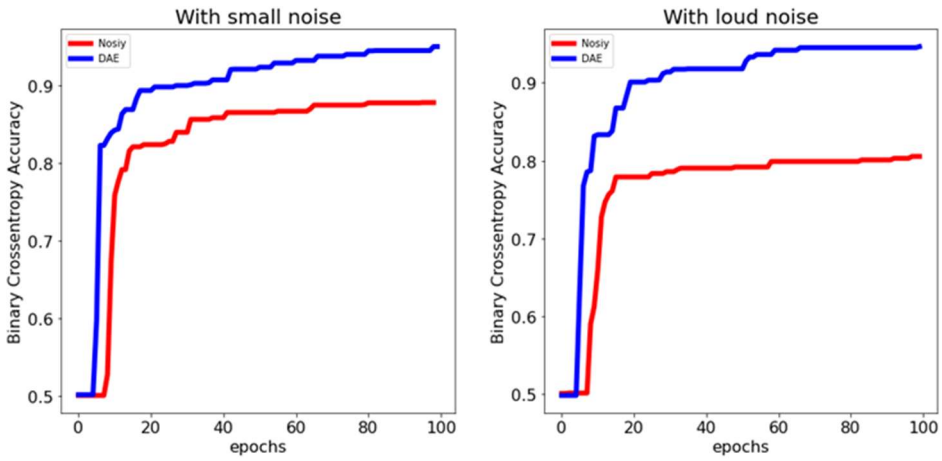


Figure 4.4. Comparison of model performance in small and loud noise environments.

4.2. Case 2 – Manufacturing industry for small quantity batch production

Another place where this system is demonstrated is a small quantity batch production factory using several types of manufacturing equipment such as a hydraulic oil press, welding robot, and various conventional machines described in Figure 4.5. We targeted manual milling, a hydraulic oil press, and a manual lathe.

To obtain the training dataset, we controlled the environment so that ambient noise could not be heard to obtain training dataset. Then, we recorded various combinations of target machines without any control to verify the model. The noise at this place is louder and more varied than in the lab environment. The sounds include living noise and non-target equipment such as a hand drill, hoist, and grinder.

We optimized the preprocessing parameters of this system through the same process applied in the laboratory environment. Figure 4.6 shows the results of the performance by different preprocessing parameters. As a result, the frame length, the window length ratio, and the filter bank were determined to be 0.5 seconds, 0.1, and Log-Mel type, respectively.

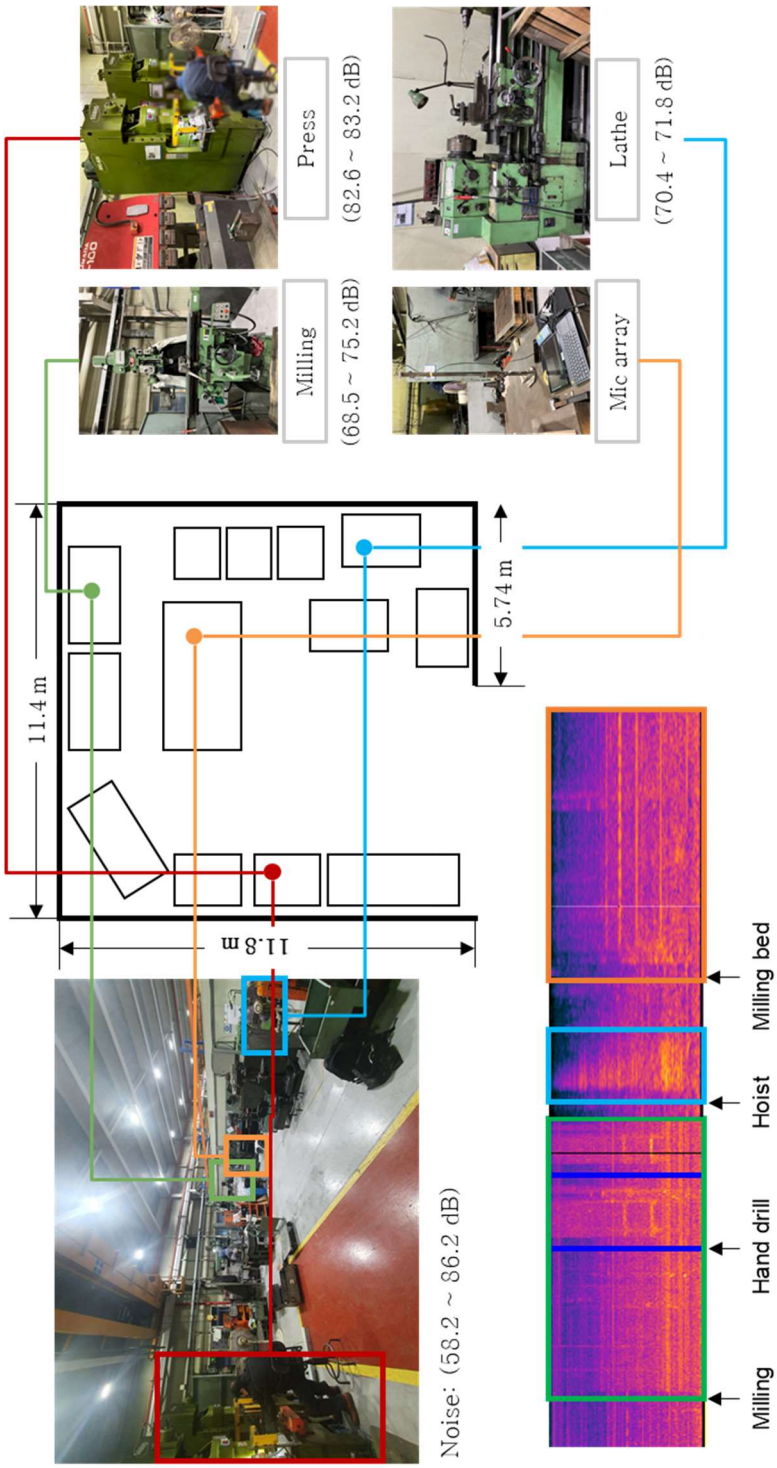
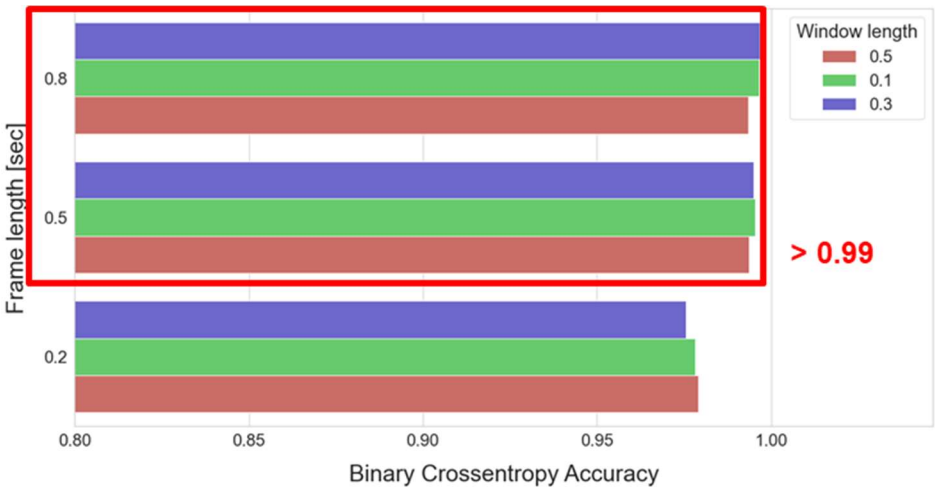
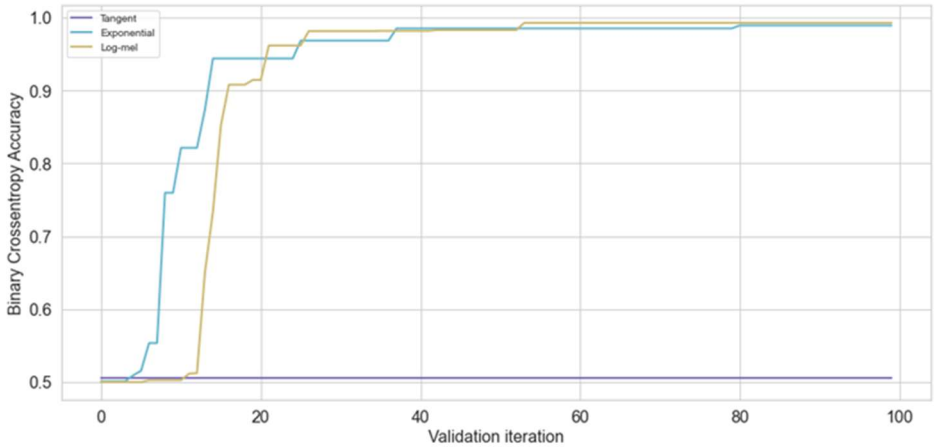


Figure 4.5. Layout of sound monitoring system and the target devices in the manufacturing site and a spectrogram of noise



(a)



(b)

Figure 4.6. Comparison of binary cross entropy accuracy of each preprocessing parameter. (a) binary cross entropy accuracy by frame lengths and window lengths (b) binary cross entropy accuracy by filter banks.

We trained the classification model to consider the noisy environment with generated data from an autoencoder. Then, we finally evaluated the monitoring system at the factory without any control. The operation monitoring results for three pieces of equipment are shown in Figure 4.7. When the lathe was operating alone or in conjunction with the milling, this system inferred the operating state with almost 100% accuracy. On the other hand, the monitoring accuracy for the simultaneous operation of the press and the milling was about 72%, and the performance was relatively poor. Overall, the sound monitoring model infers the operation of the equipment with a 93% accuracy and 0.92 F1 scores in the noisy manufacturing area.

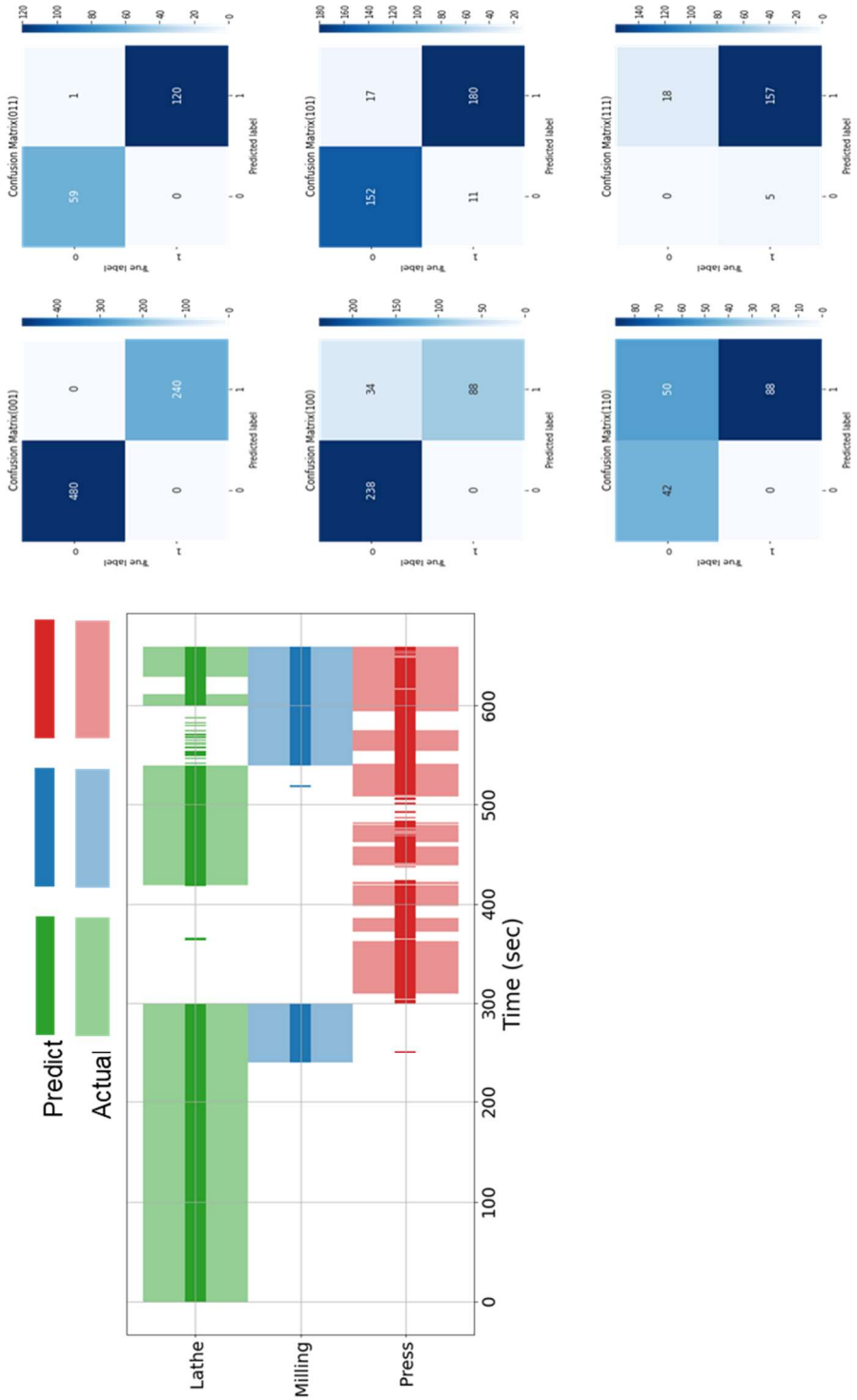


Figure 4.7. Process monitoring results at the manufacturing site

Chapter 5. Conclusion

In this study, we developed a sound monitoring system that can classify several machine sounds with low computational cost but in real-time and with high accuracy so that the system can be embedded in the edge computer. We analyzed the machine sound using STFT and converted it to a spectrogram with optimized preprocessing parameters. We also designed the DSC-based classification model to be suitable for edge computing systems by reducing the computational cost. In addition, we applied an autoencoder network for data generation that minimizes the impact of noise to build the system even in noisy environments robustly. Consequently, the model performed classification in a laboratory environment with a 94% accuracy in a noisy environment with optimized preprocessing parameters. While in a manufacturing area, it classified the operation state of each machine with a 93% accuracy with 1.1 seconds inference time.

With the sound-based real-time monitoring system, manufacturing industries can access information obtained from legacy machines and reduce the burden of constructing the network infrastructure because the edge computers transmit only processed data, not raw data. Moreover, if this system is developed to specify a particular location using phase difference through a mic array, it will apply to factories with multiple identical pieces of equipment.

Bibliography

[1] C. Bai, P. Dallasega, G. Orzes, and J. Sarkis, "Industry 4.0 technologies assessment: A sustainability perspective," International Journal of Production Economics, vol. 229, p. 107776, 2020.

[2] C. Bai, S. Kusi-Sarpong, and J. Sarkis, "An implementation path for green information technology systems in the Ghanaian mining industry," Journal of Cleaner Production, vol. 164, pp. 1105-1123, 2017.

[3] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," IEEE Access, vol. 6, pp. 6505-6519, 2017.

[4] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, "The expected contribution of Industry 4.0 technologies for industrial performance," International Journal of Production Economics, vol. 204, pp. 383-394, 2018.

[5] D. Griggs, M. S. Smith, O. Gaffney, J. Rockstrom, M. C. Ohman, P. Shyamsundar, W. Steffen, G. Glaser, N. Kanie and I. Noble, "Sustainable development goals for people and planet," Nature, vol. 495, no. 7441, pp. 305-307, 2013.

[6] S. Kolla, M. Minufekr, and P. Plapper, "Deriving essential components of lean and industry 4.0 assessment model for manufacturing SMEs," Procedia CIRP, vol. 81, pp. 753-758, 2019.

[7] M. Prause, "Challenges of industry 4.0 technology adoption for

SMEs: the case of Japan," Sustainability, vol. 11, no. 20, p. 5807, 2019.

[8] W.-K. Jung, D.-R. Kim, H.-S. Lee, T.-H. Lee, I.-S. Yang, B.-D. Youn, D. Zontar, M. Brockmann, C. Brecher, and S.-H. Ahn, "Appropriate smart factory for SMEs: concept, application and perspective," International Journal of Precision Engineering and Manufacturing, vol. 22, no. 1, pp. 201-215, 2021.

[9] P. D. U. Coronado, R. Lynn, W. Louhichi, M. Parto, E. Wescoat, and T. Kurfess, "Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system," Journal of Manufacturing Systems, vol. 48, pp. 25-33, 2018.

[10] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," Manufacturing Letters, vol. 3, pp. 18-23, 2015.

[11] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems-Reviews, methodology and applications," Mechanical Systems and Signal Processing, vol. 42, no. 1-2, pp. 314-334, 2014.

[12] G. Niu, B.-S. Yang, and M. Pecht, "Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance," Reliability Engineering & System Safety, vol. 95, no. 7, pp. 786-796, 2010.

[13] G.-Y. Lee, M. Kim, Y.-J. Quan, M.-S. Kim, J.-Y. Kim, H.-S.

Yoon, S. Min, D.-H. Kim, J.-W. Mun, J.W. Oh, I.G. Choi, C.-S. Kim, W.-S. Chu, J.K. Yang, B. Bhandari, C.-M. Lee, J.-B. Ihn, and S.-H. Ahn , "Machine health management in smart factory: A review," Journal of Mechanical Science and Technology, vol. 32, no. 3, pp. 987-1009, 2018.

[14] H. Kim, W.-K. Jung, I.-G. Choi, and S.-H. Ahn, "A low-cost vision-based monitoring of computer numerical control (CNC) machine tools for small and medium-sized enterprises (SMEs)," Sensors, vol. 19, no. 20, p. 4506, 2019.

[15] W.-K. Jung, H. Kim, Y.-C. Park, J.-W. Lee, and S.-H. Ahn, "Smart sewing work measurement system using IoT-based power monitoring device and approximation algorithm," International Journal of Production Research, vol. 58, no. 20, pp. 6202-6216, 2020.

[16] J. Kim, H. Lee, S. Jeong, and S.-H. Ahn, "Sound-based remote real-time multi-device operational monitoring system using a convolutional neural network (CNN)," Journal of Manufacturing Systems, vol. 58, pp. 431-441, 2021.

[17] Y. Ma, Y. Zeng, and S. Sun, "A software defined radio based multi-function radar for IoT applications," in 2018 24th Asia-Pacific Conference on Communications (APCC): IEEE, pp. 239-244, 2018.

[18] J. K. Sinha and K. Elbhbah, "A future possibility of vibration based condition monitoring of rotating machines," Mechanical Systems and Signal Processing, vol. 34, no. 1-2, pp. 231-240, 2013.

[19] E. Gadelmawla, "Computer vision algorithms for measurement and inspection of external screw threads," Measurement, vol. 100, pp. 36–49, 2017.

[20] G. Tristo, G. Bissacco, A. Lebar, and J. Valentinčič, "Real time power consumption monitoring for energy efficiency analysis in micro EDM milling," The International Journal of Advanced Manufacturing Technology, vol. 78, no. 9, pp. 1511–1521, 2015.

[21] J. Chen and X. Ran, "Deep learning with edge computing: A review," Proceedings of the IEEE, vol. 107, no. 8, pp. 1655–1674, 2019.

[22] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[23] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," IEEE Access, vol. 6, pp. 6900–6919, 2017.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[25] S. Bhowmick, S. Nagarajaiah, and A. Veeraraghavan, "Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos," Sensors, vol. 20, no. 21, p. 6299, 2020.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[28] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition," Speech Communication, vol. 108, pp. 15-32, 2019.

[29] H.-C. Li, Z.-Y. Deng, and H.-H. Chiang, "Lightweight and resource-constrained learning network for face recognition with performance optimization," Sensors, vol. 20, no. 21, p. 6114, 2020.

[30] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

Abstract

데이터 활용에 대한 중요성이 대두되면서, 산업현장은 다양한 센서의 유의미한 정보를 통한 스마트 공장화를 지향하고 있다. 스마트 공장화를 위한 첫번째 단계는 공장에서 발생하는 영상, 전류, 소리 등 데이터를 수집하고 이를 통해 기계 장비를 모니터링하는 것이다. 많은 종류의 센서 중, 소리 데이터는 현장의 많은 정보를 담고 있을 뿐 아니라, 쉽게 획득할 수 있는 경제적인 데이터이다. 하지만 소리 데이터 원본은 비교적 용량이 크다는 점과 개인정보 보호의 어려움으로 인해 중앙 데이터 센터에서 다루기에는 제약사항이 많다. 이를 해결하기 위해, 본 연구에서는 에지 컴퓨터를 기반으로 소리 데이터 분석을 통한 실시간 장비 가동 모니터링 시스템을 제안한다. 제안하는 시스템은 에지 컴퓨터 역할을 수행하는 Jetson Nano와 데이터 취득을 위한 마이크 어레이 Respeaker Mic Array v2.0로 구성된다. 이 시스템을 이용하여 공작기계로부터 발생하는 소리를 녹음한 후, 무작위 혼합, 소리의 진폭 조정 등을 통해 데이터를 증강하였으며, 신호 처리 구간에서 모델에 적합한 최적의 파라미터를 선정하였다. 또한, 에지 컴퓨팅을 위해 작지만 빠른 추론이 가능하며, 높은 정확도를 갖는 모델을 개발하였다. 실제 공장에서 발생할 수 있는 노이즈에 대한 영향을 고려하여 Autoencoder 기법이 사용되었다. 이 시스템은 시제품을 제작하는 연구실 장비들을 이용하여 개발되고 검증되었으며, 실제 소품종 대량생산을 하는 제조업체에 적용되었다. 결과적으로 이 시스템은 93%의 정확도와 1.1초의 추론 시간으로 장비들의 작동상태를 모니터링하였다.