Hierarchical Flow-Based Anomaly Detection Model for Motor Gearbox Defect Detection

Younghwa Lee[1], Il-Sik Chang[1], Suseong Oh[2], Youngjin Nam[2], Youngteuk Chae[3], Geonyoung Choi[3], and Gooman Park[4*]

[1] Department of Information Technology and Media Engineering, The Graduate School of Nano Design Fusion, Seoul National University of Science and Technology, Seoul 01811, Korea
[e-mail: younghwaya@seoultech.ac.kr, e-mail: ischang@hanmail.net]

[2] IT Media Engineering Program, Department of IT and Media Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea
[e-mail: oss415@seoultech.ac.kr, issuure@g.seoultech.ac.kr]

[3] Advanced R&D Department, SMR Automotive Modules Korea Ltd, 116 Jomaru-ro 427beon-gil, Bucheon-si, Gyeonggi 14556, Korea
[e-mail: youngteuk.chae@motherson.com, geonyoung.choi@smr-automotive.com]

[4*] Department of IT and Media Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea
[e-mail: gmpark@seoultech.ac.kr]
*Corresponding author: Gooman Park

Received October 12, 2022; revised November 18, 2022; revised January 6, 2023; revised March 16, 2023; revised April 11, 2023; accepted June 5, 2023; published June 30, 2023

Abstract

In this paper, a motor gearbox fault-detection system based on a hierarchical flow-based model is proposed. The proposed system is used for the anomaly detection of a motion sound-based actuator module. The proposed flow-based model, which is a generative model, learns by directly modeling a data distribution function. As the objective function is the maximum likelihood value of the input data, the training is stable and simple to use for anomaly detection. The operation sound of a car's side-view mirror motor is converted into a Mel-spectrogram image, consisting of a folding signal and an unfolding signal, and used as training data in this experiment. The proposed system is composed of an encoder and a decoder. The data extracted from the layer of the pretrained feature extractor are used as the decoder input data in the encoder. This information is used in the decoder by performing an interlayer cross-scale convolution operation. The experimental results indicate that the context information of various dimensions extracted from the interlayer hierarchical data improves the defect detection accuracy. This paper is notable because it uses acoustic data and a normalizing flow model to detect outliers based on the features of experimental data.

Keywords: Motor gearbox, Operating sound, Mel-spectrogram, Anomaly Detection, Normalizing Flow, Hierarchical feature extraction.

1. Introduction

Vehicles are made up of many different parts. If any of these components fail, the vehicle quality and performance will degrade. Defects in small parts may result in part failures, which, in turn, lead to significant financial loss, vehicle accidents, and even loss of lives. Therefore, the detection of defected parts is highly important. By exploiting the advances in artificial intelligence technology, various deep-learning-based defect detection technologies, which provide accurate and uniform defect detection, have been investigated. However, the occurrence frequency of abnormal samples is significantly lower than that of normal samples due to the nature of the manufacturing process. Also, the types of defects are diverse, and it is difficult to obtain training data. As a result, the detection problem is difficult to be approached as a classification problem based on supervised learning. Consequently, the manufacturing industry uses anomaly detection (AD). AD can efficiently learn from imbalanced data to distinguish between normal and abnormal parts [1].

AD (or outlier detection (OD)) is the problem of identifying patterns in data that do not conform to an expected behavior [2]. It is a method for developing a model, which is based on training data. These data have different features than the existing data. This model identifies the data rather than the noise. OD has been applied to various fields, such as the internet of things (IoT) [3-4], defect detection of industrial equipment, medical diagnosis [5-8], and abnormal image analysis (CCTV image) [9].

There are several research methods for AD. These include a network training method for reconstructing the characteristics of normal data [10], One-Class Classification Method [11-12], a feature matching method for distinguishing anomalies based on the feature distance or the probability distribution [13], and a method for directly predicting the probability values of test data using normalizing flow [14].

Among them, the AD technique, which uses normalizing flow, employs a generative flow model. By inversely transforming the continuous probability distribution function, the distribution of data can be obtained, and data loss becomes minimal. Furthermore, since the maximum likelihood of input data is used as the objective function, it is simple to use these data for AD.

In this study, we use training data obtained from the sound of a small-actuator module installed in an automobile side-view mirror. Currently, during the manufacturing process, tests are conducted in a booth, where the noise is blocked by a person directly listening to the sound. Defects can be classified using this process. However, the difference between normal and defective operation sounds obtained from the small-actuator module is very small. Consequently, different inspectors may not be able to distinguish these sounds, not even an experienced inspector. The training data used in this study have the drawback that the ground truth is not clear because of the different ways humans distinguish normal from abnormal signals and classify them. In this study, the features of the training data are examined, and an outlier identification model based on these features is proposed. The operation sound of the small-actuator module is converted into a Mel-spectrogram image and used as training data. A generative model based OD system using normalizing flow is proposed.

The proposed system is composed of an encoder and a decoder. Image data of various sizes are used as an input to the pretrained feature extractor in the encoder. Then, from the middle layer of the feature extractor, features of varying sizes are hierarchically extracted and used as input data in the decoder.

The contributions of this study are as follows:

- The training data used in this study are the operation sound signals of a small-actuator module of an automobile side-view mirror.
- The characteristics of the sound data collected for the experiment are examined and preprocessed. An AD method, which is based on analytical features, is also proposed. This method uses a novel flow-based generative model.
- In the layering stage of the feature extractor, the AD performance can be improved using the hierarchically extracted features and the size variation of the input image.

This paper is organized as follows. In section 2, related studies, such as the change of variables theorem, the concept of normalizing flow, and the flow-based generative model, are presented. The structure of the proposed model, which consists of an encoder and a decoder, is described in section 3. The characteristics and preprocessing methods used in the experiment for the operation sound data of the side-view mirror motor are described in section 4. In section 5, the performance evaluation results of the proposed AD model are presented. The experimental results, limitations, and future challenges of the proposed AD model are summarized in section 6.

2. Related studies

2.1 Change of Variables Theorem

Flow-based generative models estimate probability distributions based on the normalizing flow technique (described in subsection 2.2), which is performed by employing variable cleanup changes. The change of variable theorem is a method used to simplify problems, where the original variable is replaced by a function of another variable or multiple variables.

$$f: R^d \rightarrow R^d \quad , \quad Y = f(X), \quad X = f^{-1}(Y) \tag{1}$$

$$P_Y(y) = P_X\left(f^{-1}(y)\right)\left|det\frac{df^{-1}}{dy}\right| = P_x(x)\left|\det\left(\frac{df}{dx}\right)^{-1}\right| = P_x(x)\left|det\frac{df}{dx}\right|^{-1} \tag{2}$$

$$logP_Y(y) = logP_x(x) - log\left|det\frac{df}{dx}\right| \tag{3}$$

When an invertible function f is defined as in Equation (1), the probability distribution for the probability variable Y can be transformed into a probability distribution for the probability variable X as in Equation (2).
By introducing the logarithmic function to both sides of Equation (2), a change of variables can be derived, as shown in Equation (3).

f is a function from $R^d \rightarrow R^d$. R^d denotes a d-dimensional range. P_X represents the probability distribution for the random variable x, and P_Y represents the probability distribution for the random variable y.

2.2 Normalizing Flow

The flow-based generative model applies the invertible function $f_i(\cdot)$ to the latent variable z to model the random variable for the given data x by employing the change of variables theorem. In other words, it is possible to model complex probability distributions by calculating f with inverse functions in any easy-to-find distribution z.

$$x = z_K = f_K \circ f_{K-1} \circ \ldots \circ f_1(z_0) \tag{4}$$

$$logP(x) = logP_K(z_K) = logP_0(z_0) - \sum_{i=1}^{K} log \left| det \frac{\partial f_i}{\partial z_{i-1}} \right| \tag{5}$$

As shown in Equation (4), the transformation process of the latent variable $z_{i=} f_i(z_{i-1})$ is called the flow. Also, as shown in Equation (5), the entire variable transformation process for modeling the data x is performed. Collectively, it is defined as normalizing flow. During the training of the flow-based generative model, the process maximizes $logP(x)$ in Equation (5). Also, the process of calculating the Jacobian determinant of the variable transformation function $f_i(\cdot)$ is included. If the Jacobian determinant becomes complex, the computation load increases, and the computation speed decreases. Therefore, the variable invertible function $f_i(\cdot)$ is modeled in a form that is easy to obtain the Jacobian determinant.
z_K represents k latent variables, $f_k(\cdot)$ means the reversible transform function when $i = k$. And $P(x)$ is probability distribution.

2.3 Flow-based generative model

A flow-based deep-learning generative model directly models the probability distribution $P(x)$. That is, the objective function of the model, $L(D)$ can be obtained as a negative log-likelihood for the training data D, as shown below:

$$L(D) = -\frac{1}{|D|} \sum_{x \in D} logP(x) \tag{6}$$

To calculate Equation (6), *P(x)* must be transformed into an inverse function, and a variable transformation function of a form that is easy to obtain the Jacobian determinant must be formulated. For this purpose, the additive coupling layer, which is the most basic form of the bipartite flow series generative model [15], was proposed. The artificial neural network is learned by stacking layers, as shown in Equation (7). The part converted by $m(\cdot)$is crossed layer by layer to enable the modeling of all the dimensions of data X, where $m(\cdot)$ means a complex function, and x_1 and x_2 are values obtained by splitting the input X.

$$L \begin{cases} y_{1=} x_1 \\ y_{2=} x_2 + m(x_1) \end{cases} \quad \Leftrightarrow \quad \begin{cases} x_{1=} y_1 \\ x_{2=} y_2 - m(y_1) \end{cases} \tag{7}$$

In [16], an affine coupling layer that divides D-dimensional data X into x_1, x_2 and computes them was represented.

$$x_1 \in R^d, \quad x_1 \in R^{D-d}, \quad s,t : R^d \rightarrow R^{D-d}$$

$$\begin{cases} y_1 = x_1 \\ y_2 = x_2 \odot \exp\big(s(x_1)\big) + t(x_1) \end{cases} \tag{8}$$

$$\begin{cases} x_1 = y_1 \\ x_2 = (y_2 - t(y_1)) \odot \exp\left(-s(y_1)\right) \end{cases} \tag{9}$$

$$J = \begin{bmatrix} I_d & o \\ \dfrac{\partial y_2}{\partial x_1} & diag(\exp(s(x_1))) \end{bmatrix} \tag{10}$$

s and t are learned using an artificial neural network, and the transformation function is defined by Equation (8). Therefore, the Jacobian determinant is expressed in the form of a lower trigonometric function, and the matrix operation of a triangular matrix can be easily calculated by multiplying the diagonal matrix.

s and t stand for scale, translation and are functions from $R^d \rightarrow R^{D-d}(d < D)$. $\odot$ represents element-wise product, I_d means a d-dimensional identity matrix.

3. Hierarchical Anomaly Detection Model

As shown in **Fig. 1**, the proposed AD model consists of an encoder, which extracts input data features and a decoder, which is a normalizing flow part.

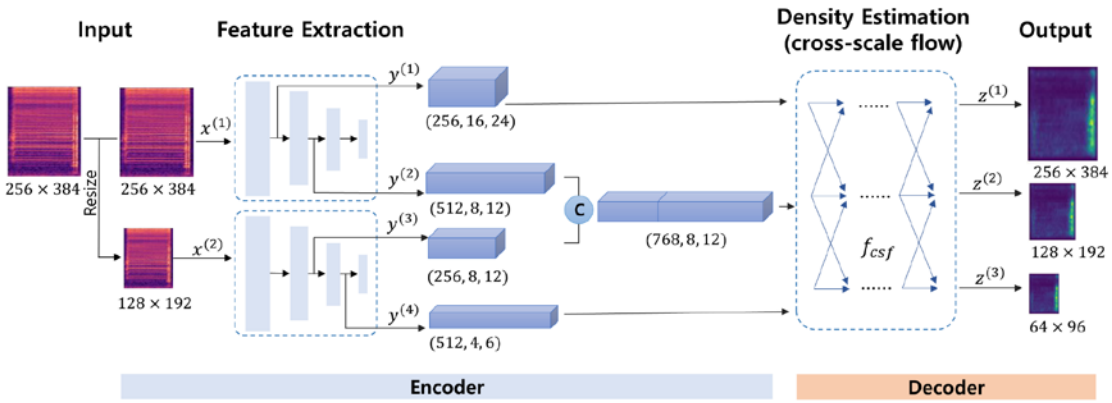

Fig. 1. Overview of the proposed hierarchical anomaly detection model

3.1 Encoder

In this study, the feature extractor is a model based on a convolution neural network (CNN). This model was pretrained using ImageNet [17]. A receptive field is an efficient feature of a CNN-based encoder. Since abnormal data come in a variety of sizes and shapes that are not standardized, they must be processed using a receptive field. A CNN-based feature extractor is important in this process. The feature map has high resolution and low-level features such as edges, curves, and straight lines, close to the input layer. On the other hand, the feature map obtained from a deeper layer has low resolution and extracts high-level features, which can infer a class such as texture, pattern, or part of an object. As shown in **Fig. 1**, in this paper, the data are extracted from the middle layer by varying the feature size. These data are used as input data. The images used as encoder input data have the original size and 1/2 of the original size. These two images are fed into each feature extractor, and two features of varying sizes are hierarchically extracted from the intermediate layer of the feature extractor. Using channel concatenation, the second feature extracted from the original size image and the first feature

extracted from the original ½-size image are combined into one feature. The decoder receives three features of varying sizes as inputs.

3.2 Decoder

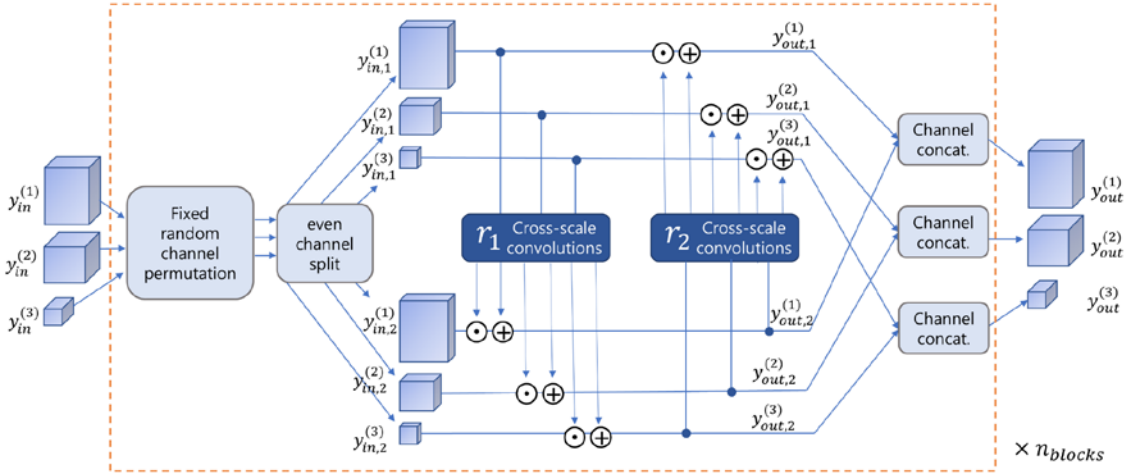

Fig. 2. Architecture of one block inside the cross-scale flow

In this paper, the cross-scale flow is used to process feature maps of different sizes that interact with each other. The cross-scale-flow method performs an extended affine transformation by employing the real valued non-volume preserving transformation (Real NVP) architecture [16] based on the coupling layer introduced in subsection 2.3. It internally divides each input tensor $y_{in}^{(i)}$ equally into $y_{in,1}^{(i)}$ and $y_{in,2}^{(i)}$. This part calculates $y_{out,1}^{(i)}$, $y_{out,2}^{(i)}$ by regressing the element-wise scale and the shift parameters applied successively to each of them to obtain the output. As shown in **Fig. 2**, the element-wise scale and shift parameters are estimated by combining the individual subnetwork $r_1$and r_2 of the blocks, which are divided into $[s_1,\ t_1]$ and $[s_2,\ t_2]$. These can be expressed together, as shown in Equation (11), where s_1 and s_2 are scale parameters, and t_1 and t_2 are shift parameters. And $\odot$ represents element-wise product.

$$\begin{aligned} y_{out,2} &= y_{in,2} \odot e^{r_1 s_1(y_{in,1})} + r_1 t_1(y_{in,1}) \\ y_{out,1} &= y_{in,1} \odot e^{r_2 s_2(y_{out,2})} + r_2 t_2(y_{out,2}) \end{aligned} \tag{11}$$

4. Data analysis

In this study, the operation sound signals of a car's side-view mirror motor were used as the experimental data. The external environmental sound may act as a hindrance factor in the analysis of the motion sound. Therefore, the recording was carried out in a recording studio equipped with soundproofing facilities to obtain accurate experimental data. The motor was fixed using a jig made to record the operation sound of the side-view mirror motor, and the acquired audio signal was saved in a wave format using a self-developed Python program. The motor consists of a normal motor and three types of abnormal motors. The motor sound has a total operation time of 7 seconds, including folding (3 seconds), unfolding (3 seconds), and an operation standby (1 second). The distinction between normal and defective motor gearboxes used in the experiment was determined by an experienced inspector working on the actual

production line by directly listening to the operation sound. As shown in **Table 1**, a defect identification number was assigned according to the type of defect.

Mel spectrograms of normal data and three types of abnormal data are shown in **Fig. 3**. **Fig. 3** (a) shows the normal motor operation sound. Shaft misalignment noise, also known as Fault 1, is a bad rotational fricative noise caused by the rotation shaft of a gear not being precisely aligned. This is a 'buzzing' noise, as illustrated in **Fig. 3** (b).

Tooth scratch noise, classified as Fault 2, is a periodic bad friction noise caused by scratches in the slope of the toothed gear during rotation. Usually, it only happens once, when folding or unfolding the mirror. However, it can happen twice, when there are dents or ruptures in the toothed gear teeth. It can be identified as a periodic 'tick-tock' pattern, resulting in a regular noise form, as illustrated in **Fig. 3** (c).

Shaft thrust noise, also known as Fault 7, is a thrusting sound produced when a motor shaft hits the motor wall. A single 'click' noise is generated at the start of folding or unfolding, and it has a different pattern than the normal sound, as shown in **Fig. 3** (d). The motion sound was saved in a wave file using a 44.1-kHz sampling rate and converted into a Mel-spectrogram image [18] using an audio library (torchaudio) of PyTorch. The training data were divided into folding and unfolding motion sounds. The option value of torchaudio was set to n_fft = 4096, win_length = 2048, hop_length = 250, and n_mels = 700 to create an image file size of 162 × 375 pixels. The generated image files consisted of 327 normal data and 100 abnormal data, 228 normal data for training and 99 normal data, and 100 abnormal data for testing.

Table 1. Fault numbers of different noise types

Fault No.	Type of Noise
1	Shaft misalignment noise
2	Tooth scratch noise
3	Shaft thrust noise

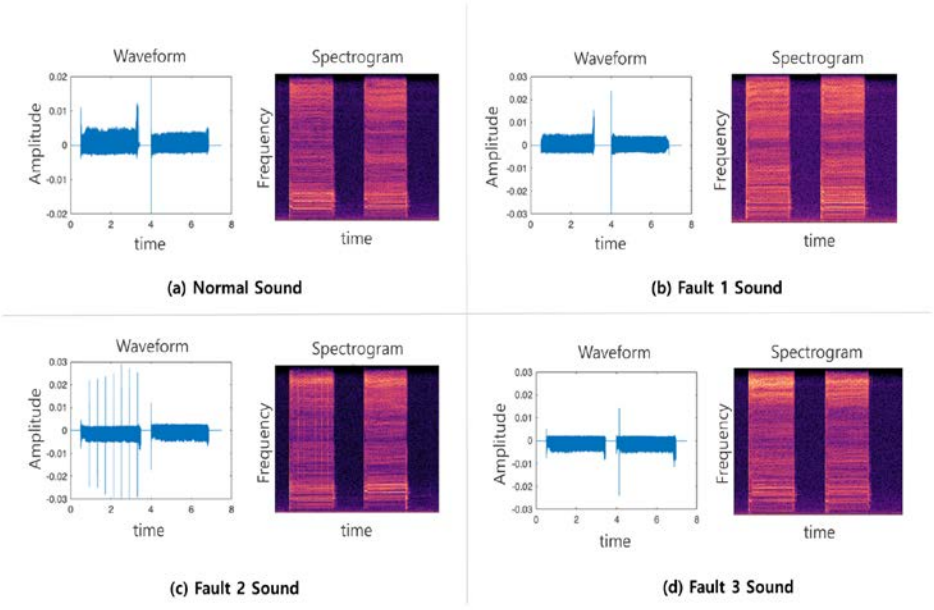

Fig. 3. Waveform and Mel-spectrogram of car's side view mirror operation sound, (a) Normal sound; (b) Fault 1 sound; (c) Fault 2 sound; (d) Fault 3 sound.

5. Experiments

5.1 Experimental Setup

The data used in the experiment were resized to 256 × 384 without cropping. The area-under-the-receiver operating characteristic (AUROC) curve was used as an evaluation metric. The cross-scale flow included four blocks; the optimizer used was Adam [19], the learning rate was set to 2×10^{-4}, the weight decay was set to 10^{-5}, the batch size was set to 64, and the epoch was set to 240.

The pretrained feature extractors used were Resnet 18 [20], WideResnet50 [21], MobileNetv3_Large [22], and MobileNetv3_Small [22]. The experiment was conducted three times, and the average values of the experimental results were obtained.

5.2 Experimental Results

5.2.1 Results according to the input data type

To examine the effect of the hierarchical data extraction method on the performance of the defect detection system in the proposed model, three types of input data were used in the experiments.

These data types can be classified according to the CS-Flow [23] method; a multiscale type with three different sizes of the input image, a multifeature type, which hierarchically extracts features of different sizes from the middle layer when a single-size data input passes through a feature extractor, and a hybrid type, which is a combination of the multiscale type and the multifeature type. The sizes used for the multiscale model were set to 256 × 384 pixels, 128 × 192 pixels, and 64 × 96 pixels. The feature sizes used in the multifeature model were (32, 48), (16, 24), and (8, 12). The hybrid type input size was used by extracting features of (16, 24), (8, 12), and (4, 6) from 256 × 384 pixels and 128 × 192 pixels sized images.

As shown in **Table 2**, the experimental results showed that the model that uses the hybrid type input data (by employing ResNet18 as a backbone) exhibits the highest performance. We observed that the performance of this model can be improved by employing the multifeature type rather than the multiscale type. However, by employing only the multifeature type, some performance degradation was observed compared to the hybrid type.

Table 2. AUROC of our dataset for the proposed model according to the input data type and feature extractor

Feature extractor	**Types of input data**		
	multiscale (CS-flow)	multifeature	**hybrid (Ours)**
ResNet18	0.922	0.951	**0.958**
WideResNet50	0.88	0.908	0.885
MobileNetV3_L	0.923	0.940	0.943
MobileNetV3_S	0.873	0.921	0.951

A histogram illustrating the distribution of testing data according to the type of data used (multifeature, multiscale, and hybrid) using ResNet18 as a backbone is shown in **Fig. 4**.

Fig. 4 (a) (multiscale type) shows that both the abnormal and normal data are widely spread, resulting in many overlapping parts. On the other hand, **Fig. 4** (b) (multifeature type) shows that both the normal and abnormal data are well concentrated on one side. **Fig. 4** (c) shows

that the distribution of normal data is narrowly saturated, and some overlapping sections are observed. However, both the normal and abnormal data are more widely separated than those in **Fig. 4** (a), indicating that various outliers have been identified well.

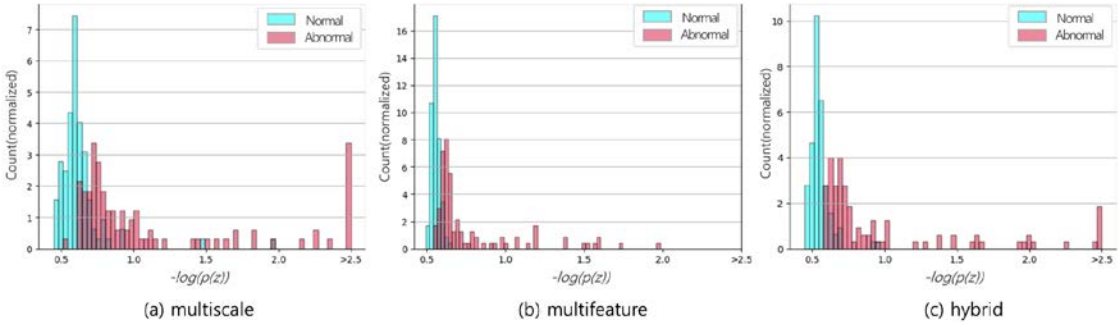

Fig. 4. Distribution histogram according to types of input data

Reducing the size of input data is similar to the pooling process in the CNN standard. This is like the subsampling process. As the parameters are reduced, the representation of the network is also reduced. Thus, overfitting can be suppressed. However, this may reduce the network confidence. Therefore, by employing only the multiscale type, the amount of information in the data decreases, and the network representation also decreases. Consequently, the data distribution diverges because the characteristics cannot be clearly expressed in the generated data. However, by employing the multifeature type, relatively few pooling processes are required because features are hierarchically selected at the intermediate stage of the layer. Therefore, various levels of low-dimensional and high-dimensional information are obtained, so that it is possible to learn both local and global information about the data. Thus, the data tend to be distributed in a convergent direction. However, in our data, there was little difference between normal and abnormal data. By employing the multifeature type, many overlapping parts in the normal distribution and the defective distribution can be observed. Therefore, a multiscale factor was added to separate the distribution of normal and abnormal data. The size of the input data was configured differently, and features were extracted and used from the middle layer of the layer. Hierarchically extracted data has features of various dimensions, and this process improved the accuracy of defect detection.

The effect of the number of blocks on the performance was also investigated. As shown in **Table 3**, the experimental performance increases by employing up to four blocks. After that, it decreases.

Table 3. AUROC of our dataset of the proposed model according to the number of blocks inside the cross-scale flow

number of blocks	1	2	3	4	5	6
AUROC	0.931	0.951	0.957	**0.961**	0.950	0.956

5.2.2 Comparison results using MVTec AD

Table 4 shows the results obtained when applying MVTec AD [24] data to the proposed model. The encoder backbone network was set to Resnet18, and the input image size was set to 256 × 384, which is the same size used in our data images.

MVTec AD are data acquired through real industrial sensors for AD research. They contain 15 classes, 3629 pieces of training data, and 1725 pieces of testing data. Since these data are created for research purposes, there are also images produced in artificial environments such as lighting environments. The abnormal image is created based on a real environment scenario, and the ground truth is clear. In contrast, in our data, the number of training data is smaller than that of MVTec AD, and there is little difference between normal and abnormal data. Also, since the ground truth was determined by the inspector, there may be a mixture of defects in the normal training dataset itself or a mixture of normal data in the bad testing dataset. Consequently, the performance of the MVTec AD with clear ground truth was higher than that in the experimental results of our data. The reason for the performance difference is the specificity of the dataset.

However, it can be confirmed that the proposed model works well not only on our data but also on a public dataset, although its performance is not state-of-the-art (SOTA).

Table 4. Comparison of AUROC result values according to input data types applying MVTec AD to the proposed model

	Types of input data		
MVTec data type	multiscale (CS-flow)	multifeature	**hybrid (Ours)**
bottle	0.786	0.996	0.996
cable	0.5	0.961	0.945
capsule	0.866	0.920	0.9346
carpet	0.707	0.882	0.7488
grid	0.491	0.884	0.8287
hazelnut	0.5	0.959	0.9686
leather	0.687	0.999	1
metal_nut	0.5	0.967	0.9531
pill	0.591	0.920	0.8792
screw	0.504	0.993	0.8856
tile	0.708	0.968	0.9558
toothbrush	0.870	0.798	0.8583
transistor	0.538	0.904	0.9304
wood	0.592	0.984	0.9825
zipper	0.848	0.994	0.9945
Average	0.645	0.941	0.924

5.2.3 Comparison results applying our dataset to other AD models

We applied our dataset to other AD models. AD models that had performed well on research datasets, such as MVTec AD, performed poorly on our dataset. This indicates that the proposed AD model operates well, reflecting the unique characteristics of our dataset. **Table 5** shows the results.

Table 5. AUROC comparison of anomaly detection models using MVTec AD data and our data

	Dataset	
Model	**MVTec AD**	**Our data**
FastFlow [25]	0.994	0.785
CS-Flow	0.987	0.843
CFlow [26]	0.969	0.489
Differnet [27]	0.949	0.919
PaDim [28]	0.979	0.861
Cutpaste [29]	0.961	0.809
Patchcore [30]	0.991	0.909
Proposed AD model (Ours)	0.924	**0.958**

6. Conclusion

A motor gearbox defect detection system employing a hierarchical flow-based AD model was proposed. The proposed model has a network structure, in which the feature extractor extracts features of different sizes hierarchically from the layering stage, uses them as inputs to the decoder, and learns the data distribution through cross-scale flow. In this study, the operation sound of a small-actuator motor of a car's side-view mirror was used as an experimental dataset. We analyzed the training data and proposed an AD model based on the data properties. This study can be used in the design of a model based on data generated in an actual manufacturing process.

Since the performance of the SOTA model has been verified using research data, it was difficult to achieve high performance by applying it to actual data. However, we studied AD using data from the car's side-view motors as experimental data. Therefore, our results can be useful in designing fault-detection models for production lines of small motors.

In a future study, we will investigate the tradeoff relationship between input data resize and hierarchical features in the proposed AD model. If a tradeoff relationship between the two is identified, it is expected that a high-accuracy model can be built.

References

[1] Josh Patterson and Adam Gibson, "Deep Learning: A Practitioner's Approach," O'Reilly Media, Inc. 2017

[2] VARUN CHANDOLA, ARINDAM BANERJEE, VIPIN KUMAR "Anomaly Detection : A Survey," *ACM Computing Surveys*, September 2009. Article (CrossRef Link)

[3] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu and S. Yang, "LSTM-Based Analysis of Industrial IoT Equipment," *IEEE Access*, Vol.6, pp.23551-23560, 2018. Article (CrossRef Link)

[4] A. Gaddam, T. Wilkin, M. Angelova and J. Gaddam, "Detecting Sensor Faults, Anomalies and Outliers in Internet of Things: A Survey on the Challenges and Solutions," *Electronics*, Vol.9 No.3, 2020.

[5] D. Y. Oh and I. D. Yun, "Residual Error Based Anomaly Detection Using Auto-Encoder in SMD Machine Sound," *Sensors*, 18, 1308, 2018. Article (Cross Ref Link)

[6] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. endo, and Y. Kawaguchi, "Anomalous Sound Detection Based on Interpolation Deep Neural Network," in *Proc. IEEE ICASP*, 271-275, 2020. Article (Cross Ref Link)

[7] R. Lang, R. Lu, C. Zhao, H. Qin, and G. Liu, "Graph based semi-supervised one class support vector machine for detecting abnormal lung sounds," *Applied Mathematics and Computation*, Vol. 364, 124487, 2020. Article (Cross Ref Link)
[8] R. Banerjee and A. Ghose, "A Semi-Supervised Approach for Identifying Abnormal Heart Sounds Using Variational Autoencoder," in *Proc. IEEE ICASP*, 1249-1253, 2020. Article (Cross Ref Link)
[9] D. Li, D. Chen, J. Goh, S.K. Ng, "Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series," *arXiv preprint arXiv:1809.04758*, 2018. Article (Cross Ref Link)
[10] Raghavendra Chalapathy and Sanjay Chawla, "Deep Learning for Anomaly Detection: A Survey," *CoRR, (abs/1901.03407)*, 2019. Article (CrossRef Link)
[11] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys (CSUR)*, 54, 1-38, 2021. Article (Cross Ref Link)
[12] L. Ruff, R. A. Vandermeulen, L. Deecke, S. A. Siddiqui, A. Binder, E. Muller, and M. Kloft, "Deep One-Class Classification," in *Proc. of Int. Conf. on machine learning (PMLR)*, 4393-4402, 2018. Article (Cross Ref Link)
[13] Liron Bergman, Niv Cohen, Yedid Hoshen, "Deep Nearest Neighbor Anomaly Detection," *arXiv:2002.10445*, 2020. Article (Cross Ref Link)
[14] Rezende, Danilo Jimenez, and Shakir Mohamed, "Variational Inference with Normalizing Flows," *arXiv preprint arXiv:1505.05770*, 2015. Article (Cross Ref Link)
[15] Dinh, Laurent, David Krueger, and Yoshua Bengio, "Nice: Non-linear Independent Components Estimation," in *Proc. of ICLR 2015*, 2015. Article (Cross Ref Link)
[16] Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio, "Density estimation using Real NVP," *arXiv preprint arXiv:1605.08803*, 2016. Article (Cross Ref Link)
[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. of 2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009. Article (Cross Ref Link)
[18] Wongeun Oh, "Comparison of environmental sound classification performance of convolutional neural networks according to audio preprocessing methods," *The Journal of the Acoustical Society of Korea*, pp. 143-149, 31 May 2020. Article (Cross Ref Link)
[19] Diederik P Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," in *Proc. of International Conference on Learning Representations (ICLR)*, 2015. Article (Cross Ref Link)
[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. Article (Cross Ref Link)
[21] S Zagoruyko and N Komodakis, "Wide Residual Networks," in *Proc. of the British Machine Vision Conference (BMVC)*, pp. 87.1-87.12, 2016. Article (Cross Ref Link)
[22] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam, "Searching for MobileNetV3," in *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314-1324, 2019. Article (Cross Ref Link)
[23] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, Bastian Wandt, "Fully Convolutional Cross-Scale-Flows for Image-based Defect Detection," *arXiv:2110.02855*, 2021. Article (Cross Ref Link)
[24] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger, "Mvtec AD–A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, 2019. Article (Cross Ref Link)
[25] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, Liwei Wu, "FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows," *arXiv:2111.07677*, 2021. Article (Cross Ref Link)
[26] Gudovskiy, D., Ishizaka, S., and Kozuka, K, "CFLOW AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows," *arXiv preprint arXiv:2107.12571*, 2021. Article (Cross Ref Link)

[27] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn, "Same but DifferNet: Semi-Supervised Defect Detection With Normalizing Flows," in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1907-1916, 2021. Article (Cross Ref Link)
[28] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier, "PaDim: A Patch Distribution Modeling Framework for Anomaly Detection and Localization," in *Proc. of pattern Recognition, ICPR International Workshops and Challenges*, 2020. Article (Cross Ref Link)
[29] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister, "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. Article (Cross Ref Link)
[30] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Scholkop, Thomas Brox, Peter Gehler, "Towards Total Recall in Industrial Anomaly Detection," *Computer Vision and Pattern Recognition (CVPR)*, 2022. Article (Cross Ref Link)

Younghwa Lee received the M.S. degree from Seoul National University of Science and Technology, Seoul, South Korea, in 2005 and the B.S in Media Engineering from Seoul National University of Science and Technology, Seoul, South Korea, in 2002. She is currently pursuing a Ph. D. in the Department of Information Technology and Media Engineering from Seoul National University of Science and Technology.
She worked for Broadcasting graphics editor, Hyundai Home Shopping, for 20 years. She is working as a senior researcher at the IoT Convergence Research Center at Seoul National University of Science and Technology. Her research interests are deep learning, computer vision, and image processing.

Il-sik Chang received his M.S. degree from Seoul National University of Science and Technology, Seoul, South Korea, in 2011 and a B.S in Electronics Engineering from Honam University, Gwangju, South Korea, in 2001. He received a Ph.D.Candidate in the Department of Information Technology and Media Engineering from Seoul National University of Science and Technology. His main research areas are deep learning, data analysis, and image processing.

Suseong Oh is pursuing his B.S. in Electronics and IT Media Engineering from Seoul National University of Science and Technology, Korea. His research interests are Computer vision, data analysis.

Youngjin Nam is pursuing his B.S. in Electronics and IT Media Engineering from Seoul National University of Science and Technology, Korea and is expected to graduate in 2024. His research interests include deep learning, artificial intelligence, computer vision, Generative AI, voice synthesis.

Youngteuk Chae is an engineer, works for SMR Automotive Modules Korea. SMR is one of the largest manufacturers of rearview mirrors for passenger and commercial vehicles in the world. SMR also develops and provides ADAS related products and solutions. He has been engaged in the system architecture and system test of CMS(Camera Monitor System) from 2016. He received the B.E from Incheon national University in 2014.

Geonyoung Choi is a senior system engineer works for SMR Automotive Modules Korea. SMR is one of the largest manufacturers of rearview mirrors for passenger and commercial vehicles in the world. SMR also develops and provides ADAS related products and solutions. He has been engaged in the system architecture and system test of CMS(Camera Monitor System) from 2016. He received the B.E from Yonsei University in 1986, and the M.E degrees in Electronic Engineering from Korea Advanced Institute of Science and Technoloy in 1997. He worked for multimedia R&D center, Samsung Electronics, for 14 years. He was a co-founder of a venture company ICANTEK, developing security cameras. He was an adjunct professor of media IT engineering division in Seoul National University of Science and Technology. He has been engaged in the development of the multimedia processing systems over 30 years.

Goo-Man Park received a Ph.D. and a M.S. in Electronics Engineering from Yonsei University, and a B.S. in Electronics Engineering from Hankook Aviation University. He is a professor of Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology, Seoul, Korea. He was a senior engineer at Samsung Electronics, Suwon, Korea. His current research interest includes computer vision, immersive media.