VIDEO ANOMALY DETECTION FOR INDUSTRIAL EQUIPMENT WITH MULTI-PATTERN REPETITION

Takehiro Kasahara[1],Takayuki Kajiwara[2]and Hidetaka Nambo[2]

[1]Industrial Research Institute of Ishikawa, Kanazawa, Japan
[2]Kanazawa University,Kanazawa, Japan

ABSTRACT

With the growing demand for automation and efficiency in factory production, using AI anomaly detection has become increasingly important. Semi-supervised learning methods, which leverage large amounts of normal operational data, are particularly effective for this purpose. However, conventional video anomaly detection methods often struggle when applied to equipment with multiple repetitive motion patterns, generating high anomaly scores even for normal operations and leading to false positives. Moreover, while high-speed video anomaly detection systems can achieve faster processing by distributing the workload across multiple devices, they are susceptible to blind spots caused by frame loss during network transmission. To address these challenges, this study proposes a video anomaly detection method capable of accurately identifying anomalies in equipment with multiple repetitive motions and resilient to frame loss. Specifically, by taking differences in latent variable dimensions, instead of taking differences in image dimensions, it will be possible to keep anomaly scores low even for multiple patterns of operation and to indicate a high anomaly score when an anomaly occurs. This proposal can be applied not only to equipment that performs repetitive operations with a single pattern, but also to industrial equipment that performs repetitive operations with multiple patterns, such as a sorting robot that inspects goods and then places them in a basket by grade, such as grade 1 or grade 2, or a conveyor that changes delivery destinations according to destination addresses. The effectiveness of the proposed method is demonstrated through comparative evaluations with conventional approaches.

KEYWORDS

Video anomaly detection, production equipment, repetitive operations, frame lost, time information

1. INTRODUCTION

The industrial use of AI is required to improve the efficiency of production sites by utilizing digital technology such as DX, and in the tasks of product inspection and condition monitoring of manufacturing equipment, much research is being conducted to achieve uniform judgment criteria, management by quantitative and objective indicators, recording of all anomaly scores and automatic processing when anomalies occur, and cost reduction by unmanned operation. In the field of image anomaly detection, various studies have been conducted.

However, the class classification method requires the preparation of a large number of normal and anomaly data, and in some cases it is difficult to prepare a large number of anomaly data at

David C. Wyld et al. (Eds): NLAI, IBCOM, MLDS, GridCom, SCAI, ICCSEA, SPPR – 2024
pp. 209-219, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142215

actual production sites. Therefore, an anomaly detection method using unsupervised learning with image generation models, which can learn and use only a large number of normal data, has attracted attention and is expected to be used for condition monitoring of industrial equipment.

When monitoring and recording the operational status of industrial equipment, one anomaly that cannot be recognized by still image anomaly detection alone is a timing anomaly. As shown in Figure.1. Take the case where a button press robot repeats the operation A→B→C→D→A, as an example. In this case, if a snag occurs at C and the next image is also C, it is not possible to determine that a snagging anomaly has occurred only by this second single image of C. Anomalies such as an unexpected stop of the target device, a decrease in speed, or a reversal of operation, which cannot be determined from a single still image in the captured moving image and can only be detected by considering the regularity of temporal changes in the moving image, are called timing anomalies.

An easy method for detecting timing anomalies is proposed [1] and shown in Figure 2. It involves inputting videos (a series of consecutive still images) taken of the system operating under normal conditions to the AI and having it generate the next timing image. By having the AI learn in this way using many videos, the AI will be able to generate the next image of the input sequence of still images. Using this AI, the difference between the image generated by inputting a sequence of still images and the actual next image is calculated as an anomaly score, and if this anomaly score is large, it is judged to be anomaly, thereby enabling detection of timing anomalies.

However, such conventional methods have the following issues. As shown in Figure 3, assume a device that operates normally in two or more patterns, i.e., repeated operation in pattern 1 (A→C →D→A, indicated by the blue line) and repeated operation in pattern 2 (A→C→B→A, indicated by the green line). In this case, at the branchpoint of the operation patterns, the AI cannot know in advance which pattern it will operate in, so it will generate a vague image where the images of Pattern 1 and Pattern 2 are blended. In this case, whether the pattern is actually Pattern 1 or Pattern 2, both images will be different from the image generated by the AI, resulting in a large anomaly score even if the operation is normal.

In addition, when performing video anomaly detection, it is conceivable to implement an AI for anomaly detection in the cloud and send continuous images captured by the camera over the network. In this case, missing images may occasionally occur due to network instability. When missing images occur, the method shown in Fig. 2 causes the anomaly score to increase, even though no anomaly has occurred in the target device.

In this study, we propose a method for calculating an anomaly score that solves the above problem and detects timing anomalies with high accuracy in equipment that performs repetitive operations in multiple patterns, and show its properties. The condition is unsupervised learning of time-series images by an image generation model, and the accuracy of detecting timing anomalies is calculated, and the results are compared and discussed.

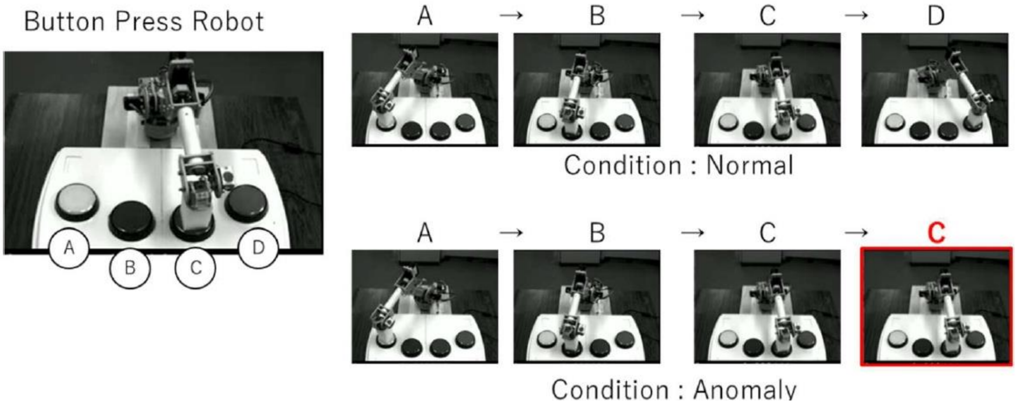


Figure 1. Examples of timing anomaly

2. Related Research

Methods to generate predictions of the next image by learning time-series images have been studied, and anomaly detection methods based on these methods have been proposed.

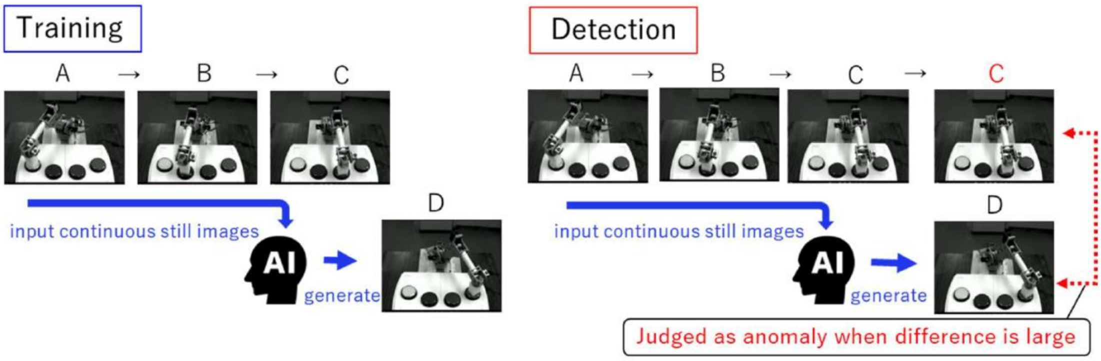

Training
A → B → C
input continuous still images
AI
generate
D
Detection
A → B → C → C
input continuous still images
AI
generate
D
Judged as anomaly when difference is large

Figure 2. Example of training and detection methods for timing anomaly detection

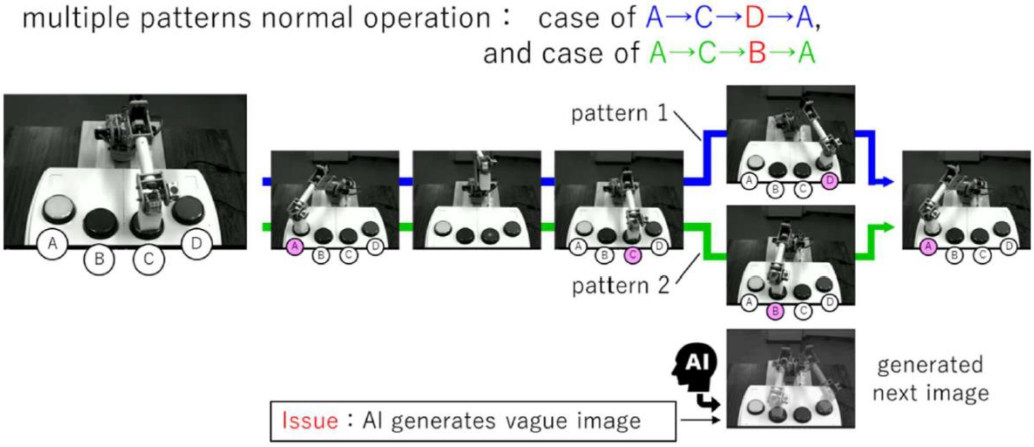

multiple patterns normal operation : case of A→C→D→A,
and case of A→C→B→A
A
B
C
D
pattern 1
pattern 2
AI
generated next image
Issue : AI generates vague image

Figure 3. Issue in the method shown in Figure 2

In a study [2], a method is proposed to predict the next image in a time-series sequence by using a hierarchical network that simulates PredictiveCoding, which has been studied as a human learning mechanism. For the rendered image, a five-layer network was constructed for a monochrome image of 64 × 64 pixels, and for the in-vehicle image dataset Kitti, a four-layer network was constructed for a color image of 128 × 160 pixels. image generation results.

In study [3], a network with multiple CNN and ConvLSTM layers was constructed for anomaly detection on time-series images, showing fast and accurate results.

In the study [4], a method was proposed to detect anomalous frames by learning from time-series images using Flownet, a network that computes optical flow. Evaluation on the CUHK Avenue dataset and the UCSD Ped dataset, which contains recordings of people walking, has shown good accuracy.

These conventional methods have not been evaluated for detecting timing anomalies in devices with multiple repetitive motion patterns.

3. Proposed Method

The purpose of this study is to propose a method for highly accurate detection of timing anomalies in video anomaly detection for devices that perform multiple patterns of repetitive motion, and to evaluate these methods to clarify their properties.

Anomaly detection methods have been proposed that use AI to generate predictions of time-series images and calculate the difference between the predicted and actual images as an anomaly score. In this method, if a device has two repetitive motion patterns, (1) Pattern 1 and (2) Pattern 2, as shown in Figure 3, and the probability of selection is 50% at each branching point, the image generation model will generate avague image where the images of Pattern 1 and Pattern 2 are blended. In this case, the difference between the generated image and the actual image is large regardless of whether the actual behavior is Pattern 1 or Pattern 2. Therefore, the anomaly score due to the difference between the generated image and the actual image becomes large even though the behavior is normal (Figure 4).

To solve this problem, we propose a mechanism that calculates a low anomaly score for normal operation and a high anomaly score for anomaly operation, even for a device that performs multiple patterns of operation, by learning using the difference of latent variables instead of only image differences in the loss function, and by using latent variables in the calculation of anomaly scores. We propose a mechanism to calculate a low anomaly score for normal operation and a high anomaly score for anomaly operation (Figure 5 left).

To cope with missing images during network transmission, we propose a mechanism to detect anomalies at arbitrary times by inputting the time information of each of the consecutive frames captured, together with the latent variables of the images, into the AI, generating images and their latent variables according to the time of the determined image, and calculating an anomaly score by comparing the actual image and latent variable pairs (Figure 5 right). This system is proposed to detect anomalies at arbitrary times.

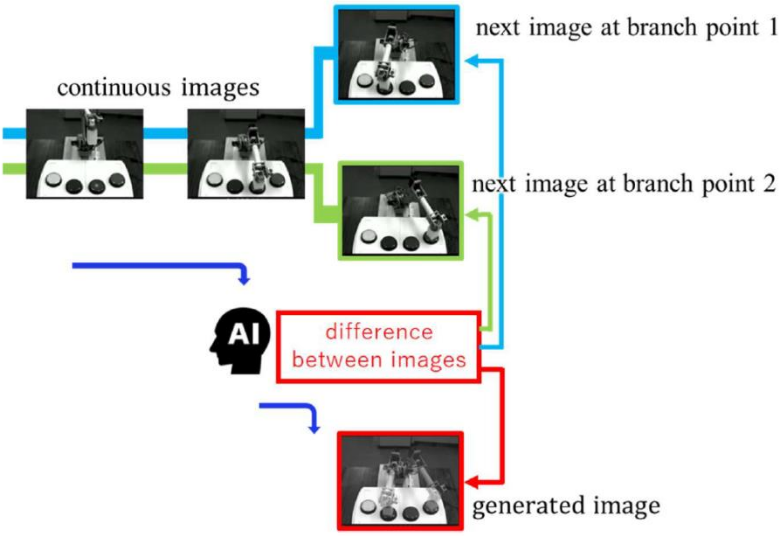


Figure 4. Conventional method.

In this paper, we evaluate the proposed method without time.As a mechanism for this, we have combined AE and LSTM to construct the proposed method (1) AELSTM as a model that incorporates the differences of latent variables in the loss function, and the proposed method (2) AELSTMdivz that employs a loss function in which the differences of latent variables z are divided and summed with unequal weights in AELSTM. AELSTMdivz is constructed and evaluated.

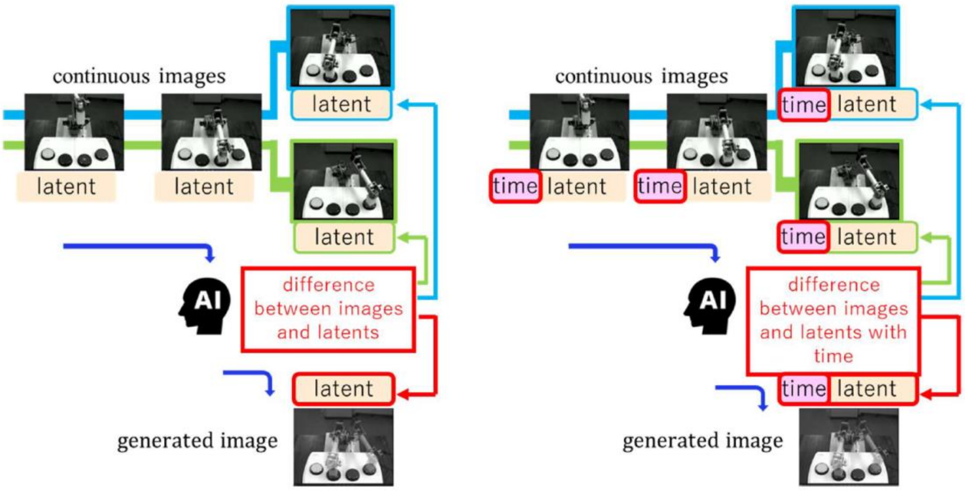


Figure 5. Overview of the proposed method without time (left) and with time (right)

In addition to these methods, we also consider a method that achieves similar effects to methods (1) and (2) by using PredNet, which has excellent image generation capability. The details of each method are described below.

3.1. AELSTM

This section describes the AELSTM model (Figure 6), which combines AutoEncoder, which encodes and decodes images, and LSTM, which can generate relationships between time series data.

Let x_t denote the image at time t, $\hat{x}_t$ the generated image, z_t and $\hat{z}_t$ the latent variables of each image, and $\mathcal{L}_{AELSTM}$ the loss function for network learning, each of which is represented by the following equations.

$$z_t = \mathrm{Encoder}(x_t) \tag{1}$$

$$\hat{z}_t = \mathrm{LSTM}(z_t, z_{t-1}, \dots, z_{t-T}) \tag{2}$$

$$\hat{x}_t = \mathrm{Decoder}(\hat{z}_t) \tag{3}$$

$$\mathcal{L}_{AELSTM} = \lambda_x \mathcal{L}_x + \lambda_z \mathcal{L}_z \tag{4}$$

$$\mathcal{L}_x = \|x_{t+1} - \hat{x}_t\|_2 \tag{5}$$

$$\mathcal{L}_z = \|z_{t+1} - \hat{z}_t\|_2 \tag{6}$$

3.2. AELSTMdivz

Among the networks that make up the AELSTM, we constructed and evaluated a network for a method in which the latent variable z is divided into N parts so that the number of elements is equal, and each part is given unequal weights for learning.

For the AELSTM network described in 3.1, the latent variable z is divided and the loss function with unequal weights is $\mathcal{L}_{AESLTMdivz}$, each of which is expressed by the following equation. Figure 7 shows an example with N=4 divisions.

$$z_t = [z_t div1, z_t div2, \dots, z_t divN] \tag{7}$$

$$\hat{z}_t = [\hat{z}div1_t div1, \hat{z}_t div2, \dots, \hat{z}_t divN] \tag{8}$$

$$\mathcal{L}_{AESLTMdivz} = \lambda_x \mathcal{L}_x + \lambda_z \mathcal{L}_z + \lambda_{zdivAll} \mathcal{L}_{zdivAll} \tag{9}$$

$$\mathcal{L}_{zdivAll} = \sum_N \lambda_{zdivN} \mathcal{L}_{zdivN} \tag{10}$$

$$\mathcal{L}_{zdivN} = \|z_{t+1} divN - \hat{z}_t divN\|_2 \tag{11}$$

Each value obtained by the loss function $\mathcal{L}$ is calculated as an anomaly score A. The following eight anomaly scores $A_{AELSTMdivz}$, A_x, A_z,$A_{zdivAll}$, A_{zdiv1}, A_{zdiv2}, A_{zdiv3}, A_{zdiv4} are calculated.

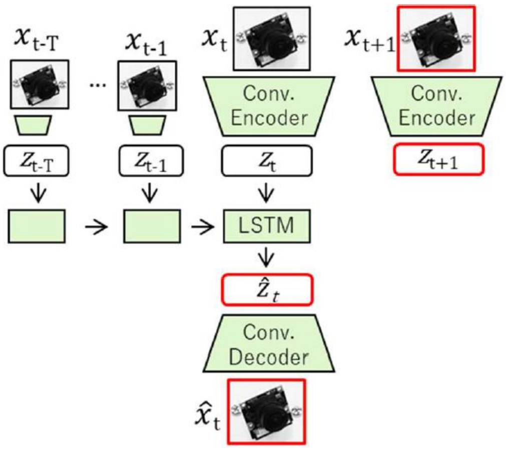


Figure 6. AELSTM network combining AE and LSTM

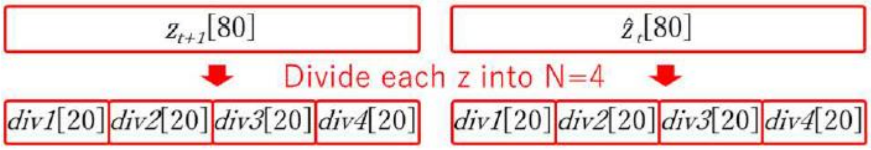

z_{t+1}[80]
$\hat{z}_t$[80]
Divide each z into N=4
div1[20] div2[20] div3[20] div4[20]
div1[20] div2[20] div3[20] div4[20]

Figure 7. Example of dividing an 80-dimensional latent variable z into N=4, four 20-dimensional latent variables

3.3. Highly Stacked PredNet

PredNet[2] has the structure shown in Figure 8 and can be stacked in any number of layers. Each component and the loss function $\mathcal{L}_{PredNet}$ are expressed by the following equation.

$$A_{l=0}^t = x^t \tag{12}$$

$$A_l^t = MaxPool\left(Relu\left(Conv(E_{l-1}^t)\right)\right) \tag{13}$$

$$\hat{A}_l^t = Relu\left(Conv(R_l^t)\right) \tag{14}$$

$$E_l^t = \left[Relu(A_l^t - \hat{A}_l^t); Relu(\hat{A}_l^t - A_l^t)\right] \tag{15}$$

$$R_l^t = ConvLSTM\left(E_l^{t-1}, R_l^{t-1}, UpSample(R_{l+1}^t)\right) \tag{16}$$

$$\mathcal{L}_{PredNet} = \sum_t \lambda_t \sum_l \frac{\lambda_t}{n_l} \sum_{n_t} E_l^t \tag{17}$$

PredNet units are stacked until the spatial size of the Representation layer features is less than or equal to the CNN kernel size. In the experiment described below, the original image reduction size in preprocessing is 96 x 96 pixels and the CNN kernel size is 3 x 3. In this case, the spatial size of the features is halved with each stacking, so six stacked layers have a side of 96, 48, 24, 12, 6, 3, which is equal to the CNN kernel size of 3 × 3. In addition to the weighted anomaly score $A_{total}$of all layers, the average value of $E_l$in the error layer is calculated as the anomaly score $A_L$in the Lth layer.

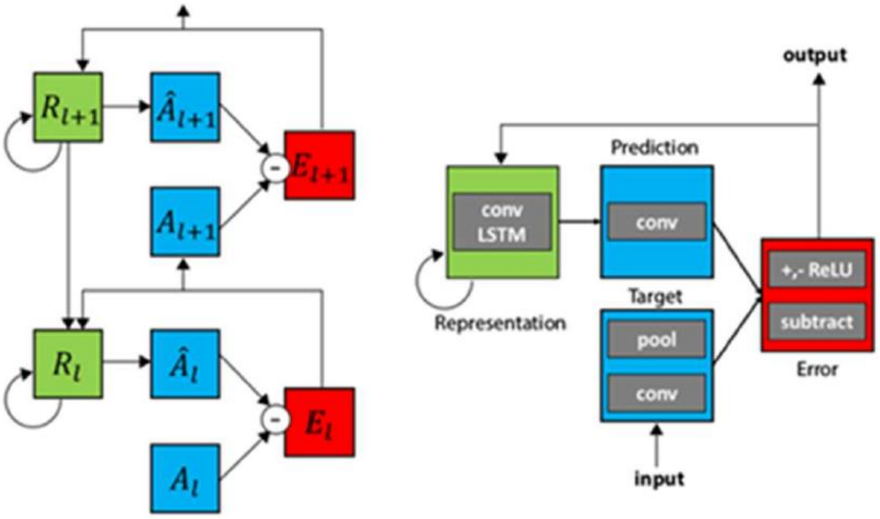


Figure 8. Stacked structure and components of PredNet [2]

4. EXPERIMENT

To evaluate the accuracy of detecting timing errors in a device that performs multiple patterns of operation, a device that operates as set as the toy problem was created, and images of this device were captured and used as a time-series image data set for evaluation. The data set was used to evaluate both the conventional and the proposed methods, and the accuracy was calculated.

4.1. Preparation of Evaluation Equipment

In order to create a time-series image data set for evaluation, we created a device that rotates an electronic component (an embedded camera component). The motor used for rotation is the Nema17 stepping motor with high time accuracy, which is pulse-controlled using the ESP-32 microcontroller and the TB6600 driver, and operated at a setting of 3200 pulses for one revolution.

4.2. Configuration of Equipment Operation

By randomly selecting one rotation to the left and stopping at the front, or one rotation to the right and stopping at the front, we set up an operation in which the rotation direction after stopping is right rotation with a probability of 50% and left rotation with a probability of 50%. Specifically, normal and anomaly motions in rotation A and B are set as shown in (1) to (4) below (Figure. 9).

(1)A. Normal rotation: One rotation to the left and stop at the front for 0.3 seconds
(2)B Normal rotation: One rotation to the right, stopping at the front for 0.3 second
(3)A.Anomaly rotation: One rotation to the left, the speed decreases to half at 3/8 to 5/8 during the rotation, and stops at the front for 0.3 seconds.
(4)B Anomaly rotation: One rotation to the right, speed drops to half at 3/8 to 5/8 of the way through the rotation, and stops for 0.3 seconds at the front.

First, either (1) or (2) is randomly selected and rotated a total of 100 times, which is assumed to be the normal operation for learning. However, (1) and (2) are selected so that they are rotated a total of 50 times each. Next, as in the training test, either (1) or (2) is randomly selected and rotated a total of 10 times, and this is the normal operation for the test. However, (1) and (2) are selected so that they are rotated a total of five times each. Finally, (3) and (4) are randomly selected and rotated a total of 10 times. However, (3) and (4) are selected so that each of them rotates a total of five times.

The average rotation speed of the normal operation is 800 pulses/second, so that one rotation is made over a period of 4 seconds. In the 5 fps imaging described below, the average number of pulses between images is 160 pulses. However, in consideration of the time variation in actual operation of industrial equipment, the pulses given to the user are also given a speed variation of 1% standard deviation. As a result, the average number of pulses between images is 160 pulses with a standard deviation of 1.6 pulses, and this standard deviation variation can also be seen in Figure 10.

4.3. Imaging Settings

A Teledyne Dalsa G3-GC10-C0640 camera was used to capture images. The image size was set to 640 x 480 pixels, the image capture mode to monochrome, the frame rate to 5 fps, and the exposure time to 10 ms. The normal operation for training was performed and the images were captured, resulting in 2149 normal images for training. The test images were taken by performing the normal operation for the test and the operation including anomalies for the test. Of these, 215 were normal images for the test and 262 were anomaly images for the test. The number of pulses between images for the test images was as shown in Figure 10.

4.4. Learning and Testing

We evaluated AELSTM, AELSTMdivz, and PredNet-L6 with 6 layers of PredNet as proposed methods, PredNet-L5 and PredNet-L4 with 5 and 4 layers of PredNet, and the method Spatio-AE by research [3] and the method Frame-Pred by research [4] as conventional methods, were evaluated.

The models were all generated by inputting nine consecutive time-series images and predicting the next image, and each anomaly score was calculated. z dimension in AELSTMdivz was set to 80, the number of divisions was N=4, and each weight was set as follows. $\lambda_{zdivAll}$ =0.1, λ_{zdiv1}=1.0, λ_{zdiv2}=0.1, λ_{zdiv3}=0.01, λ_{zdiv4}=0.001 .
The number of channels in the first six layers of PredNet is [1, 48, 96, 192, 384, 768], and the weights are set as follows.λ_t=1.0, $\lambda_{l=0}$=1.0, $\lambda_{l>0}$=0.1.

The training images were trained, and the test images were used to calculate the anomaly scores. The score calculated from 214 of the 215 normal images in the test images, excluding the 9 images that had been input in advance, was used as the normal score [OK] set, and the score calculated from 95 of the 262 anomaly images, in which the pulse rate between images decreased, was used as the anomaly score [NG] set. Among the normal images, the score calculated from the 9 images corresponding to the point where the pattern diverges between A-rotation and B-rotation after the device stops for 0.3 seconds was defined as the branch point normal score [bOK] set.

The distribution of the normal and anomaly score sets [OK-NG] was shown by ROC curves for the calculated scores, and the AUC was calculated. The distribution [bOK-NG] of the branch point normal score set and the anomaly score set was also shown by ROC curves, and AUC was calculated. The mean and standard deviation of the AUC were calculated after five trials, each using a different random number seed.

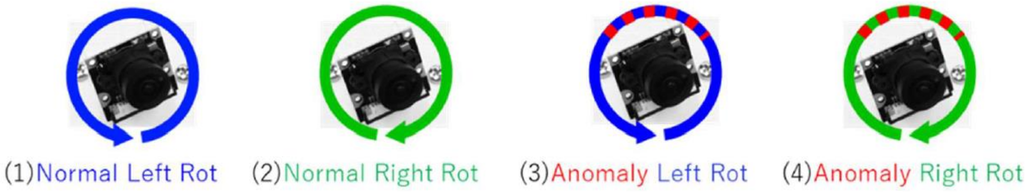


Figure 9. Normal and anomalybehavior at set rotations A and B

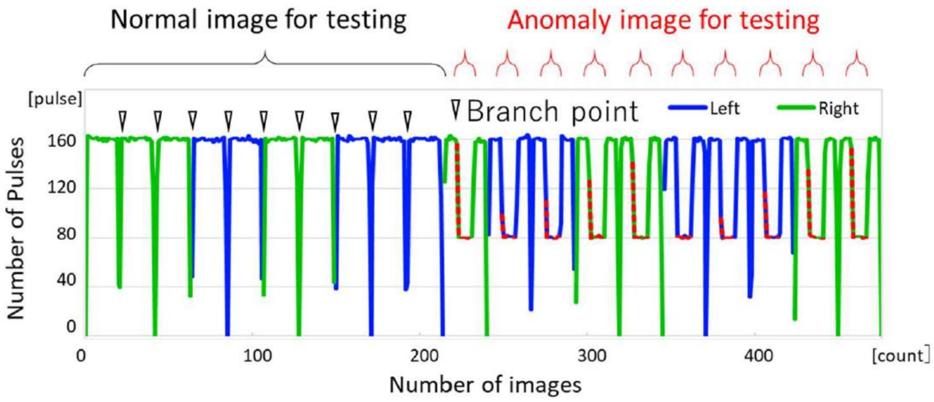


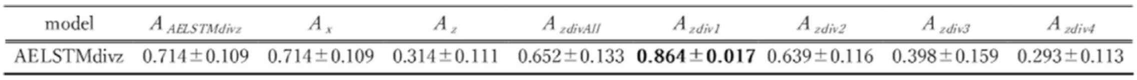
Figure 10. Number of inter-image pulses in each image

5. RESULT

Table 1 shows the AUC calculation results for the branch point normal-anomaly [bOK-NG] in each anomaly score of AELSTMdivz. The evaluation result of A_{zdiv1} showed an AUC of 0.864. Figure 11 shows the graphs of A_x, A_z,andA_{zdiv1}. In the graph, a large anomaly score is calculated in some parts of A_x, but in A_{zdiv1}, a small anomaly score is calculated in the part of the branch point normal image, and a large anomaly score is calculated in the part of the anomaly.

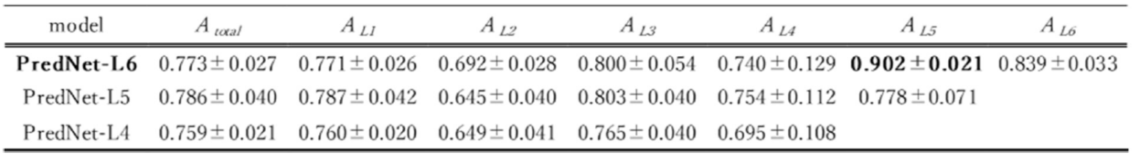
Table 2 shows the AUC calculation results for the branch point normal-to-anomaly [bOK-NG] in the various anomaly scores of the PredNet-L6, -L5, and -L4 models. The highest AUC is 0.765 and 0.803 for the number of layers 4 and 5, respectively, while the highest AUC is 0.902 (@L=5) for the number of layers 6. In addition, the normal-anomaly [OK-NG] AUC calculation showed AUCs of 0.987 (@L=1), 0.988 (@L=1), and 0.989 (@L=1) for the number of stacks of 4, 5, and 6, respectively. Figure 12 shows graphs of various anomaly scores for the PredNet-L6, -L5, and -L4 models. In the graphs, a large anomaly score is calculated for AL1 in each model at the

Table 1.AUC for each anomaly score in AELSTMdivz



model	$A_{AELSTMdivz}$	A_x	A_z	$A_{zdivAll}$	A_{zdiv1}	A_{zdiv2}	A_{zdiv3}	A_{zdiv4}
AELSTMdivz	0.714±0.109	0.714±0.109	0.314±0.111	0.652±0.133	**0.864±0.017**	0.639±0.116	0.398±0.159	0.293±0.113

Table 2.AUC for each anomaly score in PredNet-L6, -L5, -L4



model	A_{total}	A_{L1}	A_{L2}	A_{L3}	A_{L4}	A_{L5}	A_{L6}
PredNet-L6	0.773±0.027	0.771±0.026	0.692±0.028	0.800±0.054	0.740±0.129	**0.902±0.021**	0.839±0.033
PredNet-L5	0.786±0.040	0.787±0.042	0.645±0.040	0.803±0.040	0.754±0.112	0.778±0.071	
PredNet-L4	0.759±0.021	0.760±0.020	0.649±0.041	0.765±0.040	0.695±0.108		

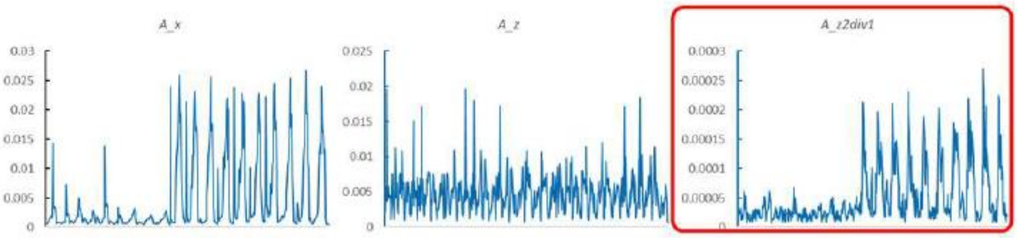


Figure 11. Anomaly Score Plot in AELSTMdivz

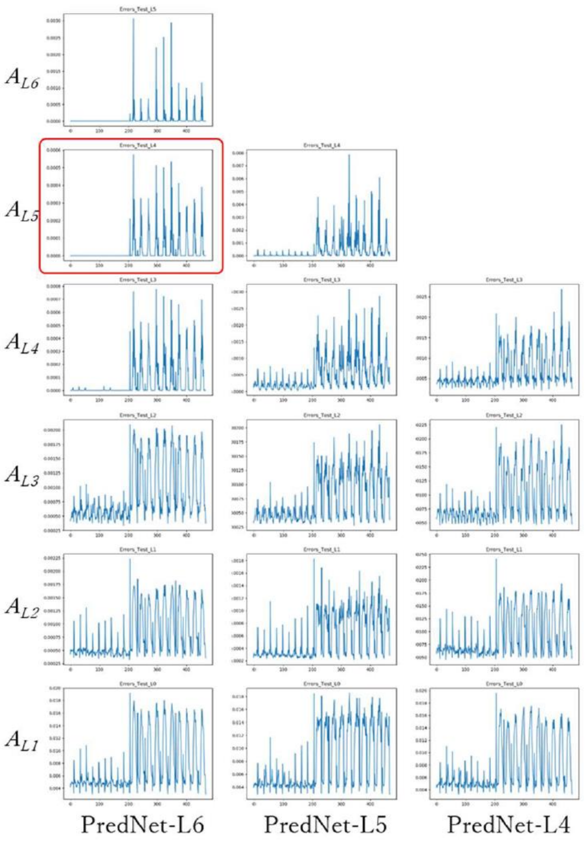


Figure 12. Anomaly score plots in PredNet-L6,-L5,-L4

position of the branchpointnormal image, while the anomaly scores at the position of the branch point normal image are suppressed to a small value for AL5 and AL6 in the PredNet-L6 model. The *Az*anomaly score in the AELSTM model and the branch point normal-anomaly [bOK-NG] in the Spatio-AE and Frame-Pred models showed AUC values of 0.729±0.017, 0.390±0.028, and 0.085±0.017, respectively.

6. CONCLUSIONS

In this paper, we focus on the problem that the anomaly score of normal operation at the pattern branch point becomes large in the detection of timing anomaly in devices operating in multiple patterns using image generation networks, and propose two methods to solve this problem. The second method is to stack PredNet features until the spatial size of the features is less than the CNN kernel size. The evaluation results show that the second method provides higher detection accuracy. This indicates that the problem of anomaly detection in repeated operation of multiple patterns may have been solved by devising a network structure. Further evaluation on a larger number of data sets will be conducted in the future.

REFERENCES

[1] Takehiro Kasahara, Taichi Nakamura, Takayuki Kajiwara and Hidetaka Nambo,: Video anomaly detection for industrial equipment that repeats multiple patterns. The 37th Annual Conference of the Japanese Society for Artificial Intelligence, 2M1-GS-10-02, 2023(in Japanease)

[2] Lotter, W., Kreiman, G., Cox, D.: Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. arXiv preprint arXiv:1605.08104

[3] Chong, S. Y., Tay, H. Y.: Anomaly Event Detection in Videos using Spatiotemporal Autoencoder. arXiv preprint arXiv:1701.01546

[4] Liu, W., Luo, W., Lian, D., Gao, S.: Future Frame Prediction for Anomaly Detection - A New Baseline. arXiv preprint arXiv:1712.0986

AUTHORS

Takehiro Kasahara is a Senior Researcher at the Industrial Research Institute of Ishikawa, Japan. He specializes in applying image recognition and deep learning to anomaly detection in industrial settings. His research focuses on identifying anomalies in image, video, and vibration data. He holds a Ph.D. in Information Science from Kanazawa University.

Takayuki Kajiwara is a master's student at Kanazawa University, where he received his B.E. in Information and Communication Engineering in 2023. His research interests include AI and machine learning.

Hidetaka Nambo is a Professor at Kanazawa University, specializing in Artificial Intelligence. He received his Doctor of Engineering from Kanazawa University in 1999. His research focuses on AI and its applications. He is a member of IEEE, IEICE, IPSJ, IEEJ and SOPEJ.