S & M ####

Development of Automatic Visual Anomaly Detection System for Data Centers

Misheel Enkhbaatar[1]* and Tatsuya Yamazaki[2]

[1]Graduate School of Science and Technology, Niigata University,
8050 Ikarashi 2-no-cho, Nishi-ku, Niigata 950-2181, Japan
[2]Faculty of Engineering, Niigata University,
8050 Ikarashi 2-no-cho, Nishi-ku, Niigata 950-2181, Japan

(Received Feburuary 6, 2024; accepted May 16, 2024)

Keywords: automatic monitoring, anomaly detection, LED, image segmentation

In this paper, we present a practical automated visual monitoring system designed to enhance the efficiency of visual inspection in data centers. Visual inspection is a manual process of detecting failures in electronic devices based on light-emitting diode (LED) lighting. The objective of data center monitoring is to implement real-time failure detection to prevent any service disruptions or loss of user data. To improve the reliability of data centers, we propose a monitoring system that automatically detects anomalies in electronic devices. The system integrates a digital camera and a novel algorithm that is tailored to distinguish normal LED lighting patterns from abnormal patterns. Experimental data were collected in an actual data center room and the system was evaluated with experiments involving LED region segmentation and anomaly detection. For the LED segmentation task, we propose a *K*-means-based method that outperformed a previous method based on background subtraction by 8%. For anomaly detection, recorded videos covering continuous monitoring of approximately 17 h were used. The proposed method successfully detected all five true anomalies in the video data. The results of another experiment for anomaly detection demonstrate that prolonged video recording for collecting patterns of LED lighting can positively contribute to a better understanding of normal patterns and can effectively be used to ensure the detection of device anomalies.

1. Introduction

The significance of data centers (DCs) has risen considerably, driven by trends such as digital transformation, cloud computing, and the Internet of Things (IoT) . DCs house a large number of electronic devices that are designed to handle processing tasks, store data, and facilitate communication with the external world.(1) Electronic devices and their environment are strictly monitored 24 h a day and 7 days a week. In the monitoring, inspectors visually check the light-emitting diode (LED) status lights of electronic devices, and this process is called visual inspection. The visual inspection method aims to ensure the proper functioning of electronic devices and has the advantage of flexibility. Adding, removing, and moving a device from one

*Corresponding author: e-mail: f21l501d@mail.cc.niigata-u.ac.jp

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

position to another is extremely frequent in DCs. Many approaches such as network-traffic-based and thermal-sensor-based monitoring methods have been proposed to automatically monitor DC devices.[2,3] Integrating monitoring target devices and the monitoring system and tuning at each physical change are compulsory in existing approaches. The installation and tuning operations result in adding extra labor costs along with daily tasks and costs to the DC operation. The visual inspection method, however, does not require such extra preparations and has the flexibility in device physical changes, which accounts for its increasing popularity.

Nevertheless, there are some challenges in the visual inspection method, including speed and consistency. The speed of performing a human visual inspection is constrained by physical ability, which results in delayed fault detection. Furthermore, the delay in anomaly detection can cause serious accidents such as service disruption and data loss. Another challenge is inconsistency in anomaly detection. Human errors including overlooking and misjudging anomalies considerably affect the consistency of visual monitoring. The challenges in the visual inspection method foster the automation of the inspection of electronic devices in DCs. The goal of this study is to develop an automation method for visual inspection that reduces bottlenecks in monitoring electronic devices in DCs.

Automatic visual monitoring is widely applied across various fields such as agriculture, manufacturing, and transportation. The integration of digital cameras and advanced image processing techniques enhances automation with high accuracy. To automate visual monitoring, two main problems should be solved. One is to adaptively parse normal or abnormal patterns from the input. Human inspectors use a reference document during visual inspection, but automatic visual monitoring must adaptively extract the reference data of what is normal and abnormal to eliminate human interaction, thereby providing speed and flexibility. Another problem of visual monitoring, especially for DCs, is to recognize LEDs from a scene as human inspectors do. Accurate LED recognition is essential for extracting lighting patterns successfully and detecting anomalies.

We surveyed several research works to understand anomalous patterns in different domains. Singh and Upadhyaya compiled various research studies related to anomaly detection with the goal of organizing knowledge across different domains and generalizing the problem definition and solution.[4] In their study, anomalies are defined as patterns in data that do not conform to a well-defined notion of normal behavior. However, they concluded that it is virtually impossible to apply techniques developed in one domain to another because the notion of anomaly significantly differs depending on the domain. Therefore, we shifted our focus to a more specific domain. Yang *et al.* surveyed the literature on classical and deep-learning-based approaches to the domain of visual anomaly detection.[5] In their paper, they mentioned that the learning system applied to anomaly detection uses knowledge from knowns such as normal data or patterns to detect unknowns. Yang *et al.* proposed that visual anomaly detection methods can be divided into two categories, supervised and unsupervised methods. Supervised visual anomaly detection is chosen when there are sufficient anomalous samples. However, in many real-world problems, anomalous samples are exceptionally infrequent and challenging to collect. Thus, in many approaches, unsupervised methods are preferred to effectively distinguish abnormal and normal patterns on an image or video dataset. The study of Yang *et al.* also reveals the effectiveness of numerous methods applied to visual anomaly detection problems. For example, deep generative models such as the variational autoencoder and others are applicable to establishing the probability distribution of images, which are considered high-dimensional data.(6–9) Despite deep generative models being applicable to image data to detect anomalies, the methods require a large number of samples and their performance is not guaranteed to be stable. Another group of methods mentioned by Yang *et al.* is composed of feature modeling methods. Instead of seeing an image as a whole, the representation of local regions of the whole image is constructed in a feature modeling method. With the feature modeling method, the dependence on the number of samples can be eliminated and it is possible to use a single sample to generate the feature distribution.(10–12)

The automatic visual monitoring of the devices in DCs should utilize an unsupervised method as in previous research because anomalous patterns of LED status appear very rarely and are unusual. Therefore, normal patterns of LEDs must be accumulated to differentiate them from anomalous patterns. To resolve this issue in DC monitoring, we initially proposed the idea of extracting a normal LED pattern from the video data recorded by a monitoring camera.(13) The extracted normal pattern was named the master data. Then, the master data was compared with subsequent videos to detect anomalies, but an excessive number of anomalies were detected in anomaly detection. There were two reasons behind this problem. One was the lack of adaptability of LED detection owing to the diversity of devices in DCs, and LED detection parameters should be adaptably specified. The other one was the insufficient video length to create the master data, because LED lighting (on/off) patterns were varied.(14–16) In this work, we defined anomalies in the electronic devices in DCs on the basis of previous studies on visual anomaly detection. Furthermore, we developed a visual anomaly monitoring system. In the system, we implemented a novel LED detection method and compared the performance of the novel method with that of the previously reported method.(13) In addition, the appropriate video length was determined on the basis of the results of experiments. The system we developed was verified in a realistic DC environment to confirm its practicality.

2. Automatic Visual Anomaly Monitoring System

2.1 System overview

Figure 1 shows the overview of the automatic visual anomaly monitoring system (AVAMS). The complete steps of determining anomalies consist of video recording and video analysis. To provide uninterrupted monitoring, a digital camera must continuously record video as a stream. Owing to physical limitations such as the shortage of memory for video buffering, we manage the video stream as separate video files at a fixed length shown as Video 1 to Video *n* in Fig. 1. The digital camera feeds AVAMS with the video files continuously recorded at the DC. AVAMS analyzes the video files received from a digital camera to detect anomalies. The analysis performed by AVAMS is divided into two main processes: LED region segmentation and anomaly detection.

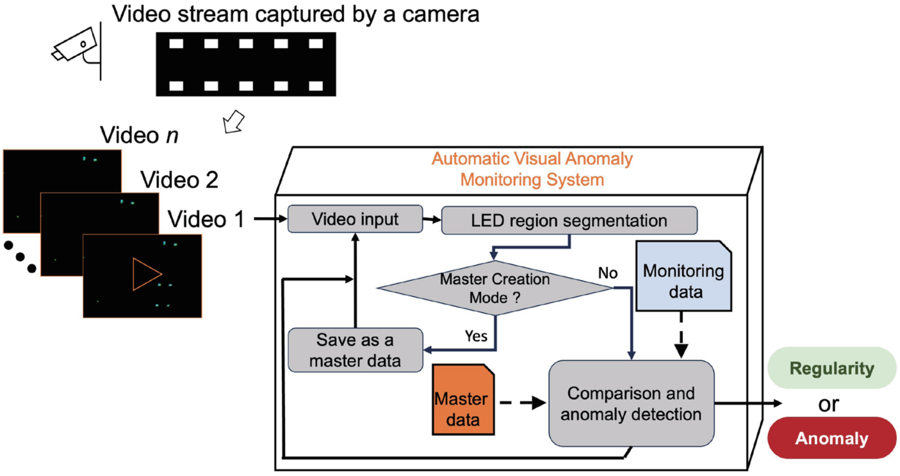


Fig. 1. (Color online) Overview of AVAMS.

The LED region segmentation process is the basis for extracting LED lighting patterns. The module of the LED region segmentation in AVAMS detects and extracts lighting LEDs on the frames of the video file as the LED regions. When the master data, which is the normal lighting pattern as mentioned in Sect. 1, is not registered, the extracted LED regions are registered as the master data. It is called the master creation mode. In AVAMS, the number of video files to be used for creating the master data is adaptively controlled by the master creation mode parameter. Namely, multiple video files can be used for creating the master data, since some LEDs may not light up on any frames in a short duration owing to LED blinking at a low frequency.

Once the master data is created in the master creation mode, the next input videos are monitored, and the same LED region segmentation module is applied to extract the LED regions, which are called the monitoring data. Then, in the comparison and the anomaly detection process, LED regions on both the master and monitoring data are compared, and anomalies are determined on the basis of the comparison. An abnormal pattern is determined when the system detects the LED that appears in the monitoring data but not in the master data. AVAMS outputs regularity when there is no difference found in the comparison and outputs anomaly otherwise.

2.2 LED region segmentation

The problem of visual anomaly detection is its requirement for adaptive and accurate image segmentation, and the LED region segmentation module plays a pivotal role in AVAMS. Image segmentation methods have been proposed in several previous reports. Deep-learning-based pixelwise segmentation methods such as U-Net and Mask Region-based Convolutional Neural Network (R-CNN) achieve state-of-the-art performance, but they require a large amount of data to annotate and train in heavy computation.(17,18) For the LED region segmentation problem, datasets are not publicly available and it is extremely difficult to collect and annotate data to train deep learning models because of the wide variety of patterns in LED lighting.

In this case, traditional image segmentation methods are more realistic for application to the problem. Because the traditional methods are known for their advantages of computational efficiency and not requiring data annotation and a large amount of sample data, they are often used in industrial applications for segmenting a moving object or detecting changes in video frames. Tarkowski *et al.* used the background segmentation method with a Gaussian mixture model (GMM) for segmenting blinking LEDs on a scene.[19] Experimental results show that the segmentation of LED regions is effective with the background segmentation method in terms of the computation of complexity in a real-life scene.

In the model proposed in a previous report,[13] background subtraction with GMM and binary thresholding methods are used to segment the LED region. The proposal of Tarkowski *et al.* considers detecting only blinking LEDs and is not sufficient for solving the anomaly detection problem. For the anomaly detection problem, both blinking and constantly lighting LED regions must be segmented from a single frame, and the pattern of lighting is extracted from sequential frames on the basis of the segmentation result. These issues urge us to design a new method optimized for anomaly detection. In the previous model, the binary thresholding method is included along with background subtraction to segment the constantly lighting LEDs. A diagram of the LED pattern extraction method used in the previous model is shown in Fig. 2. The diagram explains each step of extracting an LED lighting pattern from the input video. Each

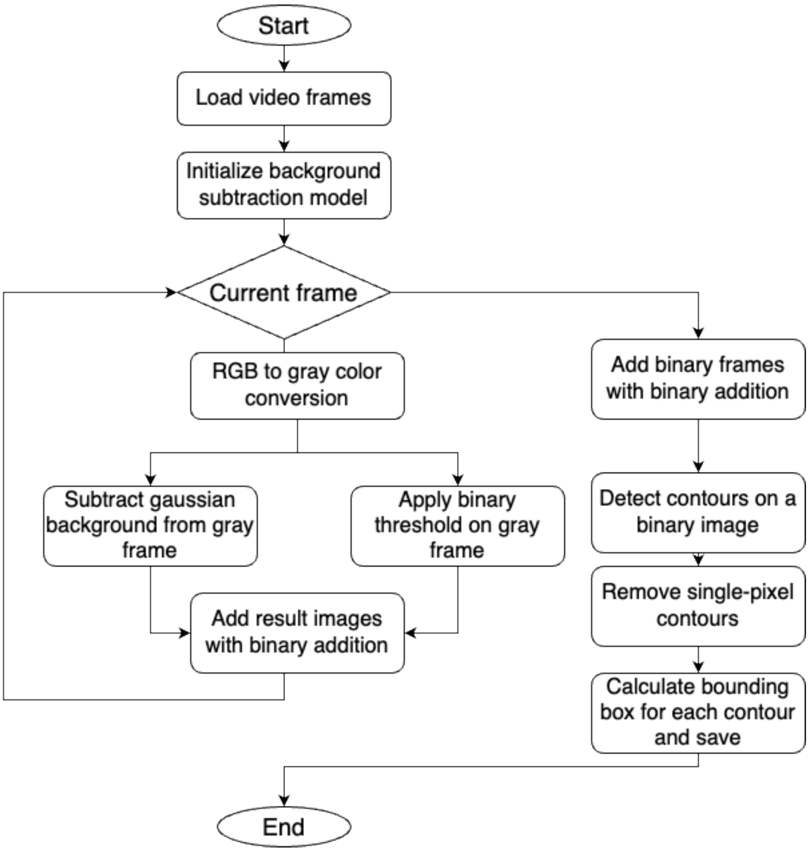

Start
Load video frames
Initialize background subtraction model
Current frame
RGB to gray color conversion
Subtract gaussian background from gray frame
Apply binary threshold on gray frame
Add result images with binary addition
Add binary frames with binary addition
Detect contours on a binary image
Remove single-pixel contours
Calculate bounding box for each contour and save
End

Fig. 2. Background subtraction and binary-threshold-based model of detecting LEDs from a video.

frame is processed separately, and the segmentation results are combined into one integrated result of the LED regions in this method. The reason of combining segmentation results is that blinking LED regions could be missed when processing video frames separately. Thus, pixelwise addition between binary frames is performed at the end of segmentation to extract the complete pattern of LED lighting. The method creates a text file containing the box (pixel position of the top left corner and width and height) information of LED regions. The file is selected as either the master or monitoring data according to the flow in Fig. 1.

However, this method developed for the previous model lacks adaptability. Complexity arises when there is a change in environment because both background subtraction and binary threshold methods require the tuning of parameters according to environmental changes. Choosing suitable parameters is extremely challenging. As mentioned in Sect. 1, the lack of adaptability caused the detection of an excessive number of anomalies that can be referred to as false positives.

In this study, we propose a novel method of segmenting LED regions. This method eliminates the extra separate handling of blinking and constantly lighting LEDs. A single-image segmentation method that relies on *K*-means clustering(20) is employed to segment a frame, thereby eliminating the need for background subtraction and binary threshold. The proposed approach utilizes binary thresholding with an adaptive threshold selection technique developed as part of this study.

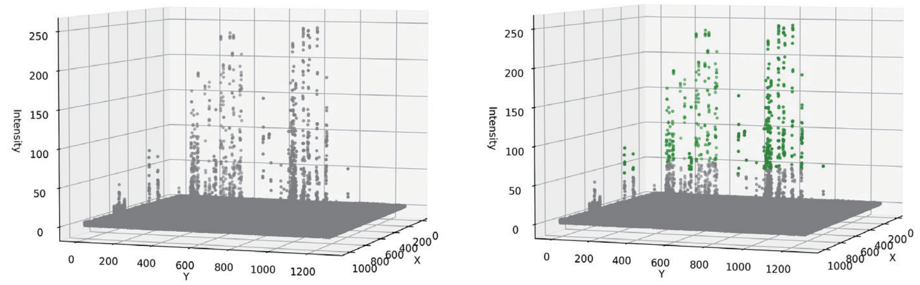
The environment of a DC room is assumed to be dark all the time except for the presence of human operators conducting a scheduled inspection or maintenance. Considering such a specific environment in the DC, brighter regions directly correspond to LED regions or the light reflection of nearby LED lights. The binary thresholding method with an optimal threshold can effectively separate the background and LED regions. Clustering image pixels into two groups (background and LED regions) by the *K*-means clustering method provides an estimation of the LED and background regions. Conventionally, it is possible to obtain pixels belonging to each cluster by calculating the Euclidean distance between cluster centers and a pixel. However, the calculation is not efficient and can enable the bottleneck in real-time systems. Instead, the threshold for binary thresholding is calculated by taking the mean of two cluster centroids and the binary thresholding outputs of the clustered image. Choosing the initial cluster centroid is challenging in *K*-means clustering. In a traditional *K*-means method, cluster centers are initiated randomly, which results in inconsistent clustering. For segmenting high-contrast pixels that correspond to the LED region and low-contrast pixels that correspond to the background region, randomly chosen initial cluster centers converge to the same cluster center and segmentation fails. The reason for this issue is that black pixels with a brightness of 0 outnumber high-contrast pixels, and the probability of choosing the same initial cluster centers of 0 brightness is very high. Furthermore, initial cluster centers may not change with the recalculation of cluster centers during the fixed iterations because nearby pixels have a brightness of 0 or an identical brightness. As a result of an additional survey to overcome this phenomenon, initial cluster centers are chosen by *K*-means++ cluster initialization, which is also an adaptive initialization technique for cluster centroids.(21) In this technique, the first cluster center is chosen randomly. Then, the second center is calculated on the basis of the probability proportional distance between the first and second centroids. The problem of selecting the same cluster of centroids can be efficiently solved with this technique because distant centroids are more likely to be selected. Segmentation by the *K*-means clustering method was ensured with sample data, and the result was illustrated in a 3D plot. The brightness of each pixel in a sample image is plotted in Fig. 3. The intensity represents the brightness of the pixel, and x and y represent the pixel position. Figure 3(a) shows the original sample image. In Fig. 3(b), the *K*-means segmentation result is illustrated with two different color values. Pixels are more concentrated within the lower region and scattered in the upper region of the plot. This means that the background region belongs to the lower part of the plot and the LED regions belong to the upper part. An adaptive segmentation with *K*-means clustering was practical for distinguishing the background and LED regions.

After the LED region segmentation, the same method we proposed in the initial AVAMS is applied. The complete steps are illustrated in Fig. 4.

3. Experimental Setup

3.1 Experimental system setting

To evaluate the effectiveness of AVAMS and the revised LED detection method, an experimental system was used to conduct necessary experiments. A digital camera and a processing machine were the essential components of the system. In the experimental setting shown in Fig. 5, we used an internet protocol (IP) camera, a processing server, and a network switch. The pair of IP camera and network switch provides video creation with customizable resolution and high-speed data transfer to the processing server. The processing server is a computational resource where all the analyses are being carried out to determine anomalies. AVAMS is deployed within the processing server. We aim to develop a computationally efficient AVAMS that does not require any high-end computer and that becomes a basic system installed in a real DC environment.



(a)

(b)

Fig. 3. (Color online) Illustration of pixel values of the sample image and K-means segmentation.

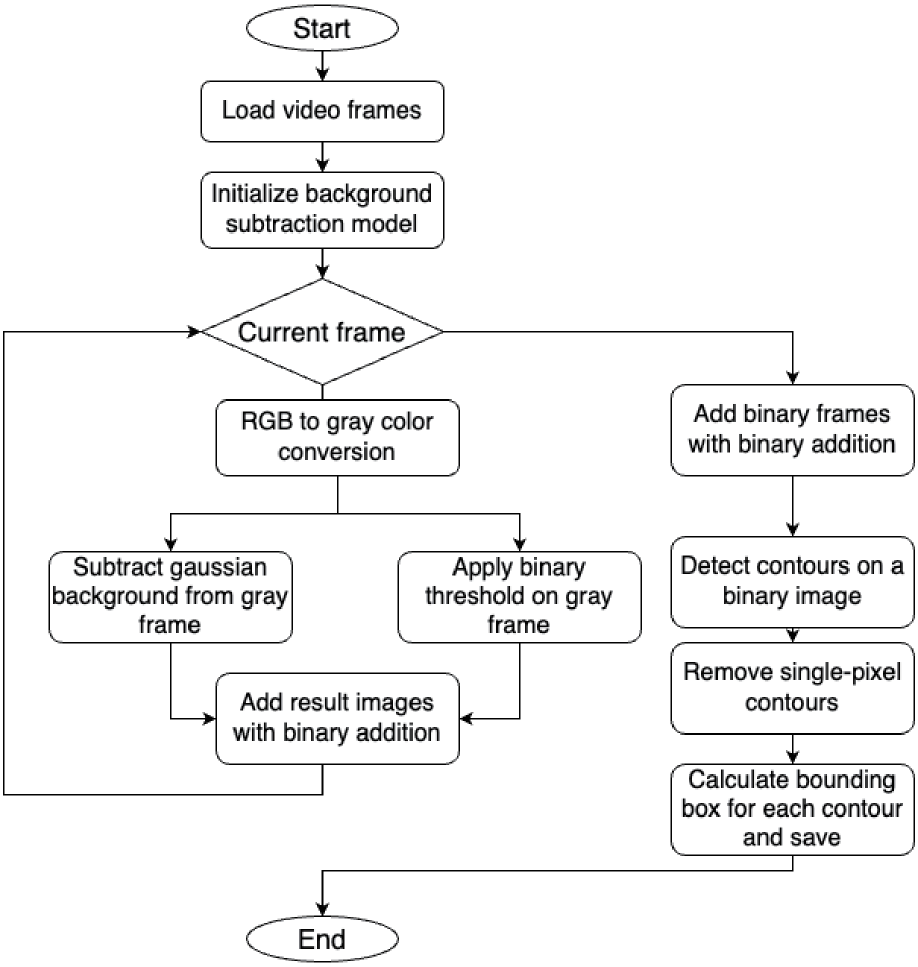

Start
Load video frames
Initialize background subtraction model
Current frame
RGB to gray color conversion
Subtract gaussian background from gray frame
Apply binary threshold on gray frame
Add result images with binary addition
Add binary frames with binary addition
Detect contours on a binary image
Remove single-pixel contours
Calculate bounding box for each contour and save
End

Fig. 4. Binary threshold with K-means-based model of detecting LEDs from a video.

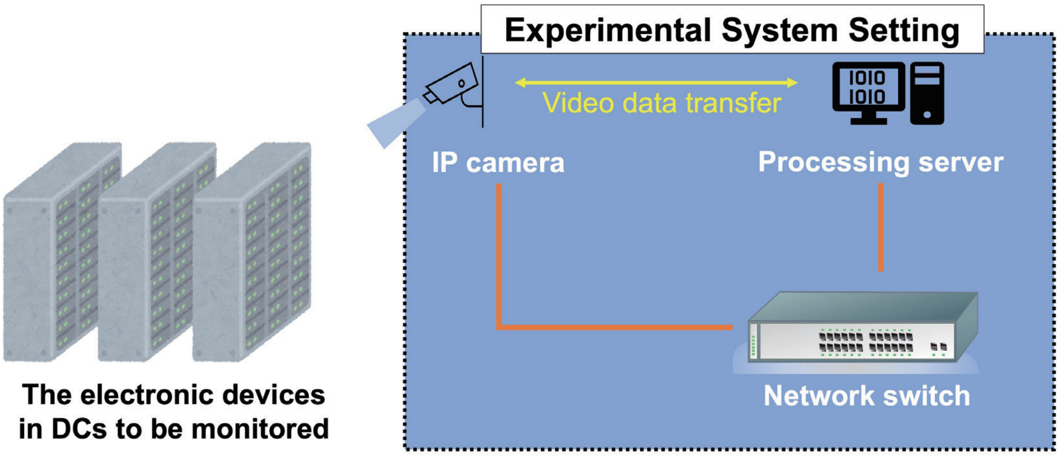

Experimental System Setting
Video data transfer
IP camera
Processing server
Network switch
The electronic devices in DCs to be monitored

Fig. 5. (Color online) Experimental system implementing AVAMS in a DC environment.

3.2 Experiment plan

To evaluate the performance of AVAMS in anomaly detection and LED region segmentation, two experiments with different datasets were planned and conducted. The first experiment corresponded to the LED region segmentation and the second to the anomaly detection by AVAMS. Datasets consist of 30-s-long videos stored in the audio video interleave (AVI) format. The frame rate of each video was 15 frames per second (fps) and the frame size was 1024×1280.

In the first experiment, the previous and novel methods were evaluated to determine whether the novel method improves the precision of LED region segmentation. The LED region segmentation experiment with the previous method shown in Fig. 2 and the novel method shown in Fig. 4 were performed on a video dataset. The evaluation metric was calculated on the basis of the comparison between manually annotated data [mostly referred to as ground truth (GT)] and the anomalies detected by the above methods. A high evaluation score was interpreted as close to human-level precision.

In the second experiment, a video length suitable for the master data creation to reduce false positives was determined. Following the video length determination, the practicality of AVAMS was ensured with the optimal video length, which was determined on the basis of the results of the analysis. The video length was represented by the number of videos. Therefore, the performance of anomaly detection was evaluated according to the video length used to create the master data in this experiment.

3.3 Data annotation

To create GT data of the LED region, bounding boxes that enclose the LED region were annotated manually. The number of LEDs, shapes, and brightness were different at each position where the videos were taken. Moreover, blinking LEDs appeared only on some frames of a video. Therefore, we split a video into frames and labeled LED regions on each frame. After that, the annotations on video frames were combined to create the final annotation. A labeling tool was used for LED region annotation, and the tool has a feature to annotate both image and video data.(22)

4. Experimental Results

4.1 Experimental situation

The experimental system was deployed in a real DC that is cooperating with this research. In the DC room, electronic devices are stored in metal racks. The IP camera used for the video recording was placed at the top of the rack to monitor devices as shown in Fig. 6. The position was carefully chosen so as not to interfere with other activities such as maintenance and regular inspection in the DC. The camera was positioned on top of the rack and monitored the racks where devices were stored, which were situated opposite to its location. The distance (d) between the camera and the recording target rack was 97 cm and the height (h) of the rack was 186.6 cm.

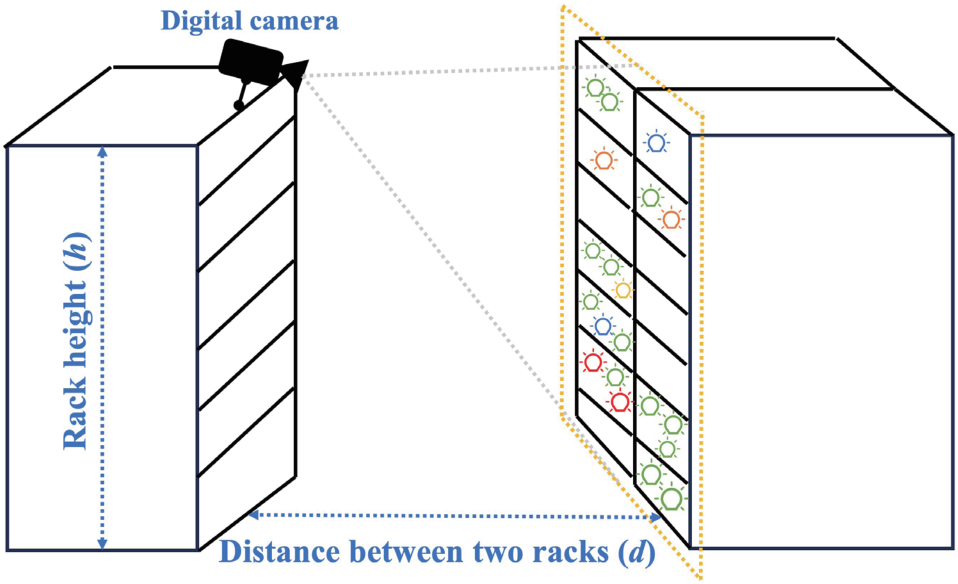


Fig. 6. (Color online) Camera position for recording videos of electronic devices in a real DC environment.

The same experimental system was brought to the DC room. The processing server and network switch were installed at the same position in one of the racks in the DC room. The camera was connected to the network switch by a local area network cable. Scenes in the room differed depending on which rack to monitor, and three different scenes were set in the experiment. The datasets of the video files were collected by three cameras installed at different positions and were labeled Datasets 1, 2, and 3, which were used for the first experiment.

Twenty videos at each position of the camera were chosen for the determination of the suitable video length in the second experiment. In practice, the total time required for completing 12 video (each 30 s long) recordings is 30 min, including video transfer and saving, and a 20 s waiting time between recordings. The number of videos at each position was chosen to be 20 to analyze for at least 30 min in the DC when we used eight videos for creating the master data. Excessive anomaly detection was observed by the previous method in a comparable period. The videos were divided into three classes according to the position from which they were taken and were labeled as Datasets 4, 5, and 6. Three hundred fifteen videos, which were more than half of 1 terabyte (TB) of a hard disk and covered approximately 17 h of monitoring, were collected at the DC to confirm the practicality of AVAMS. The videos were taken from a single position and included several abnormal patterns that were manually generated by the DC personnel. In the middle of the recordings, the DC personnel switched on the LED lighting of a device by a remote control five times at various intervals of the recording. The LED lighting was switched on at intervals in the experiment and considered an anomaly in the dataset. Note that the personnel could only provide intervals of anomalies as it was difficult to record the exact time because the switching on was by remote control. We referred to the dataset of 315 videos as the large dataset.

The processing server and the network switch were also situated in the same DC room. In the experiment, a personal computer with 11th Gen Intel i7-1185G7 3.0 GHz CPU and 16 GB of RAM was used as the processing server, and AVAMS was implemented in the processing server with C# language and OpenCV 4.0 library.

4.2 LED region segmentation experiment

Using Datasets 1, 2, and 3, we evaluated the performance of the previous method presented in Fig. 2 and the novel method presented in Fig. 4 in LED region segmentation. For the evaluation of LED region segmentation, the intersection over union (IoU) was calculated between the bounding boxes of the extracted LED region and those of GT. Namely, segmentation was judged as correct (true) when both bounding boxes overlapped beyond a threshold; otherwise, segmentation was judged as incorrect (false). In the experiment, IoU thresholds were selected from 0.25 to 0.5 at 0.05 intervals.

Then, accuracy, precision, and recall, which are the common evaluation metrics for the image segmentation task, were calculated on the basis of the true or false judgment. Definitions of accuracy, precision, and recall are given in Eqs. (1)–(3). In these equations, a true positive (*TP*) is defined as the number of bounding boxes where an LED exists both in GT and the segmentation result. A false positive (*FP*) is defined as the number of bounding boxes where an LED does not exist both in GT and the segmentation result. A false negative (*FN*) is defined as the number of bounding boxes where an LED exists in GT but no LED exists in the segmentation result, and a false positive (*FP*) is the opposite case.

$$Accuracy = \frac{TP}{TP + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Experimental results of the LED region segmentation are summarized in Table 1. Table 1 includes the results of each dataset by column. The averages of six different IoU thresholds were calculated for accuracy, precision, and recall. Each evaluation metric takes a value between 0

Table 1
Results of the LED region segmentation for each dataset.

		Dataset 1	Dataset 2	Dataset 3	Overall
Previous method	Accuracy	0.36	0.52	0.29	0.39
	Precision	0.57	0.68	0.44	0.56
	Recall	0.50	0.70	0.44	0.55
Novel method	Accuracy	0.46	0.58	0.42	**0.48**
	Precision	0.63	0.75	0.55	**0.64**
	Recall	0.62	0.71	0.64	**0.66**

and 1, and a higher value means a better result. The previous method, as well as the novel method, achieved its best performance on Dataset 2 for all metrics. The highest precision of the previous method was 0.68, whereas that of the novel method reached 0.75. In contrast, the performance of both methods dropped to the precisions of 0.44 and 0.55 on Dataset 3. In summary, the differences in accuracy, precision, and recall between the two methods were 9, 8, and 11%, respectively. Two detection samples are shown in Fig. 7 to demonstrate the results of LED region segmentation. Owing to the distance between the camera and the position of electronic devices, the height and width of LEDs are between 3 and 13 pixels. For better visualization, two samples were retrieved from full-sized images and enlarged. There are four subfigures with different captions in each sample. The position of LEDs can be observed on raw images. Yellow rectangles (bounding boxes) represent the LED regions. Annotated images show manually determined LED regions, whereas the rest of the images display detection results of the previous and novel methods. The bounding boxes visualized on the annotated image were used as GT data. In sample 1 in Fig. 7, eight LED regions were recognized and seven out of the eight LED regions were detected by the previous method. Figure 7 also illustrates the eight LED regions detected by the novel method. To demonstrate the IoU coverage of each bounding box, we selected sample 2 in Fig. 7, and IoU values are presented in Fig. 8. The segmentation results of the previous and novel methods are shown in Figs. 8(a) and 8(b), respectively. However, the minimum IoU threshold was set to 0.25 and all observable LED regions were detected by the novel method. The coverage of IoU increased in the novel method. We observed that a minor alteration in the detection of bounding boxes leads to a significant rise in IoU coverage.

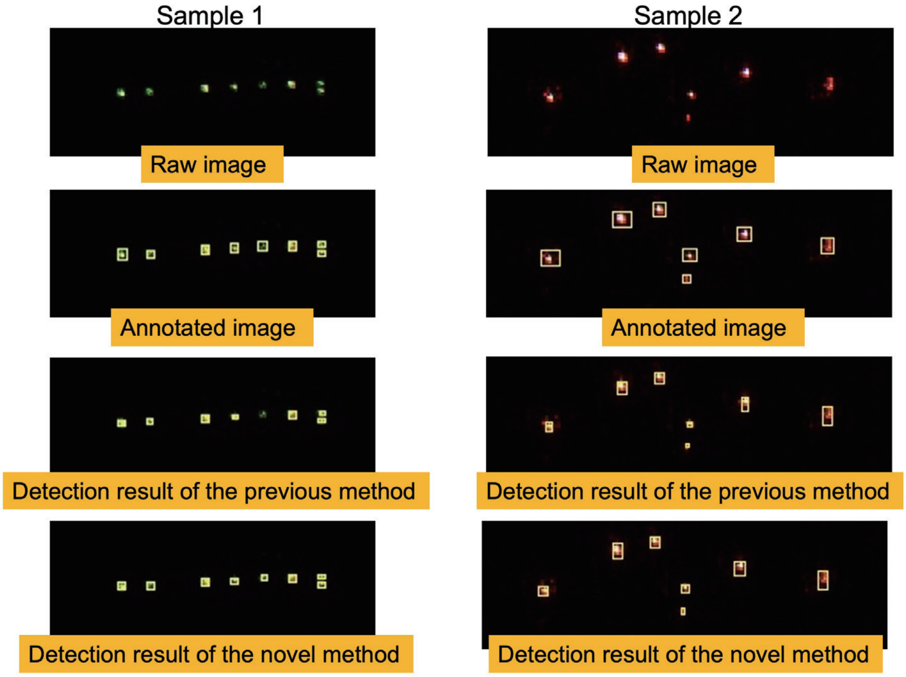

Sample 1
Sample 2
Raw image
Raw image
Annotated image
Annotated image
Detection result of the previous method
Detection result of the previous method
Detection result of the novel method
Detection result of the novel method

Fig. 7. (Color online) Comparison between annotated LED regions and segmentation results of the previous and novel methods.

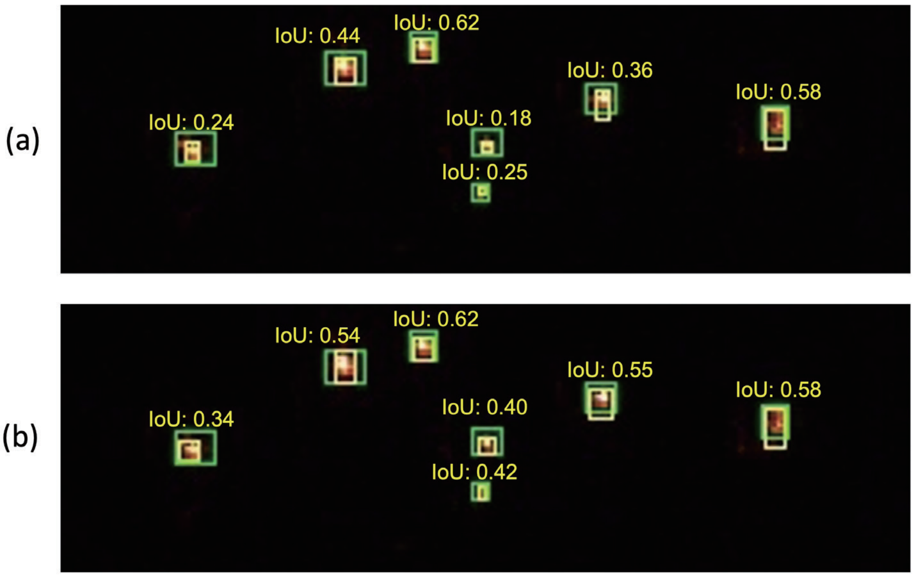


Fig. 8. (Color online) Illustration of IoU coverage of GT and detection data. A green rectangle area corresponds to GT and a yellow rectangle area corresponds to the detection result.

4.3 Video length for the master data creation experiment

The objective of the second experiment was to clarify the relationship between the video length used for the master data creation and the anomaly detection performance. In the experiment, the video length was represented by the number of videos to create the master data, and it was selected from the experimental values of 1, 4, 8, and 12. A single video duration was fixed at 30 s. Therefore, the lengths of the experimental values were 30, 120, 240, and 360 s, respectively. Datasets 4–6 were used in this experiment, and these datasets did not contain any anomalies according to the report of the DC personnel. Therefore, GT results were assumed to be all correct, and the smaller the number of detected anomalies, the more appropriate the video length for the master data creation.

An experiment result of the anomaly detection with different video lengths is demonstrated in Table 2. In each dataset, the number of incorrectly detected anomalies was counted. In general, both methods performed poorly when a video was used to create the master data. However, their performance significantly improved when the number of videos for creating the master data was increased. The novel method had only one false positive detected in the three datasets when eight sets of master data were combined. The previous method had eight false positives detected in the three datasets and 1 or more false positives detected in each dataset when eight sets of master data were combined. In the observation of combining 12 sets of master data, the novel method performed similarly to the case of combining eight sets of master data. We noted that the number of false positives detected decreased when 12 sets of master data were combined with the previous method.

Table 2
Part of results of the number of anomalies detected with different video lengths for the master data creation.

		Number of master data			
	Dataset label	(1)	(4)	(8)	(12)
Previous method	Dataset 4	11	7	3	2
	Dataset 5	19	13	1	0
	Dataset 6	19	6	4	1
Novel method	Dataset 4	4	4	1	1
	Dataset 5	18	0	0	0
	Dataset 6	3	3	0	0

Anomalies were detected in the large dataset using eight videos for the master data creation. The number of videos used for the master data was chosen on the basis of the result shown in Table 2. AVAMS detected five anomalies from the large dataset. Timestamps of anomalies that were detected during the experiment are listed in Table 3. In the experiment, the DC personnel used a personal computer that remotely controlled one of the LED lights of an electronic device in the DC room to generate artificial anomalies. The LED light remotely controlled by the personnel did not illuminate regularly. The personnel switched on the LED light at some periods to generate an artificial anomaly during the experiment and recorded the time intervals of the LEDs that actually started to illuminate as shown in Table 3. AVAMS detected illuminating LEDs that were not observed in the normal pattern. The timestamps of anomalies detected by AVAMS are within the time interval provided by the DC personnel. For the fifth anomaly, the time interval was not provided; instead, an approximated timestamp was reported. Therefore, we visually inspected all anomalies detected by AVAMS and confirmed that five anomalies originated from the sudden illumination of an LED. Additionally, AVAMS only captured five videos within those time intervals and all of them were anomalous. No false negative detection was observed in the results.

5. Discussion

In the segmentation experiment, the novel method, i.e., the *K*-means-based method, demonstrated higher accuracy, precision, and recall across all datasets. Similarly, the number of false positives in anomaly detection decreased when using the proposed method. Overall, the *K*-means-based method was considered effective in segmenting the LED regions with 8% improvement in precision than the previous method. Precision of 64% was achieved with the novel method and it was the highest precision. One of the reasons for the low precision of the LED segmentation can be the reflection of light sources observed on video frames. The light reflection results in distorted LED regions and increases the difficulty in the differentiation of separate LED regions. These challenges affect human observation and the quality of annotation. We also discovered that the segmentation depends on the LED size and the distance between the camera and the device. The server devices usually have certain LED indicator lights for power, status, and network among others, and these are more visible from the camera and have an adequate distance between LEDs. On the other hand, the network device contains LED indicator lights for each Ethernet port, and the lights are situated adjacent to each other. Figure 9 shows

Table 3
Comparison of timestamps of anomalies detected with AVAMS and generated by the DC personnel.

Index of anomalies	Anomalies detected by AVAMS (timestamp)	Anomalies generated by DC personnel (time interval)
1	2023-12-14 **16:24:30**	2023-12-14 from **16:22 to 16:24**
2	2023-12-14 **17:35:11**	2023-12-14 from **17:33 to 17:37**
3	2023-12-14 **18:58:44**	2023-12-14 from **18:58 to 19:02**
4	2023-12-14 **19:01:14**	2023-12-14 from **18:58 to 19:02**
5	2023-12-15 **08:15:07**	2023-12-15 **around 08:14**

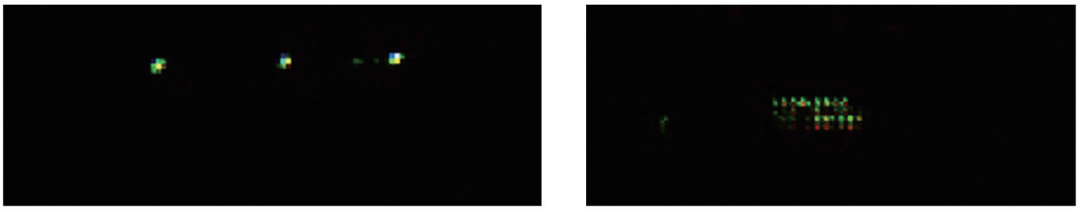


Fig. 9. (Color online) Comparison of LED groups that vary in size and distance from each other. (a) Image of group of LEDs with an adequate distance and (b) image of group of LEDs positioned in close proximity.

different types of LED group. Two sample images shown in Fig. 9 have identical heights (1.4 cm) and widths (3.5 cm). As depicted in Fig. 9, we observed that the size of LEDs and the distance between LEDs varied on two video samples. The precision of segmentation was lower on a video that included similar groups of LEDs shown in the second image of Fig. 9.

For choosing the optimal video length for creating the master data, intuitively there is a direct relationship between the video length and the correct anomaly detection. This case is not always true, because the probability of having noise in the master data increases in lengthy video data. Noise also has the same effect as light reflection. Another negative effect of noise is that abnormal patterns can be included in the master data owing to the distortion of the appearance of LEDs. This results in another excessive number of false positives detected by AVAMS. The accurate creation of the master data is the key to precise anomaly detection in electronic devices. AVAMS has a function to easily refresh the master data with a recent video. Although the master data can be refreshed until the optimal result is achieved, we observed that no more than a few refreshments were enough to obtain a result to facilitate anomaly detection. The experiment verifying the practicality of AVAMS on the large dataset showed that the number of anomalies detected by AVAMS was significantly reduced. One consideration was that AVAMS was assuming that all videos contain normal patterns. Because of the rarity of anomalies occurring, detecting all inputs as normal may show favorable performance. The successful detection of all anomalies generated manually has proven that AVAMS can indeed distinguish between normality and anomaly.

6. Conclusions

Many DCs are applying visual inspection to ensure the detection of failures in electronic devices due to flexibility. Efficiency and accuracy challenges of visual inspection are concerns for DCs gaining much attention and urging them to use an automated solution for performing visual inspection. For LED-based anomaly detection in DC devices, adaptive LED lighting pattern extraction and accurate image segmentation are the most important factors for achieving better detection results. Our proposed approach was to extract normal patterns of LED lighting. An adaptive method to collect normal patterns from one or multiple recorded videos was proposed. Next, a *K*-means-based method was proposed to segment the LED region from the videos. Experimental results indicated that the proposed segmentation method shows an 8% increase in overall precision compared with the previous method, highlighting the significance of the proposed method. Limitations included challenges in handling highly occluded objects and LEDs that are located closely together in the same location. Future work focuses on improving data annotation and extending the dataset to be evaluated.

Acknowledgments

This work was supported by JST, the establishment of a university fellowship towards the creation of science technology innovation, Grant Number JPMJFS2114.

References

1 K. Kant: Comput. Networks **53** (2009) 2939. https://doi.org/10.1016/j.comnet.2009.10.004

2 S. Zhao, M. Chandrashekar, Y. Lee, and D. Medhi: Proc. 2015 11th Int. Conf. the Design of Reliable Communication Networks (IEEE DRCN, 2015) 267–270. https://doi.org/10.1109/drcn.2015.7149025

3 Z. Chen, S. Pei, B. Tang, G. Hefferman, H. He, Q. Yang, and T. Wei: Proc. SPIE 2018 Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems. (SPIE, 2018) 88. https://doi.org/10.1117/12.2285075

4 K. Singh and S. Upadhyaya: Int. J. Comput. Sci. Issues. **9** (2012) 307.

5 J. Yang, R. Xu, Z. Qi, and Y. Shi: Procedia Comput. Sci. **199** (2022) 471. https://doi.org/10.1016/j.procs.2022.01.057

6 D. Kingma and M. Welling: Found. Trends Mach. Learn. **12** (2019) 307. https://doi.org/10.1561/2200000056

7 L. Dinh, J. Sohl-Dickstein, and S. Bengio: Proc. 2017 5th Int. Conf. Learning Representation. (ICLR, 2017). https://doi.org/10.48550/arXiv.1605.08803

8 N. Akoury and A. Nguyen: arXiv (2017). https://doi.org/10.48550/arXiv.1712.00714

9 D. Kingma and P. Dhariwal: Proc. 32nd Conf. Neural Information Processing Systems. (NeurIPS, 2018). https://doi.org/10.48550/arXiv.1807.03039

10 X. Xie and M. Mirmehdi: IEEE Trans. Pattern Anal. Mach. Intell. **29** (2007) 1454. https://doi.org/10.1109/TPAMI.2007.1038

11 T. Böttger and M. Ulrich: Pattern Recognit Image Anal. **26** (2016) 88. https://doi.org/10.1134/s1054661816010053

12 P. Napoletano, F. Piccoli, and R. Schettini: Sensors **18** (2018). https://doi.org/10.3390/s18010209

13 M. Enkhbaatar and T. Yamazaki: Proc. 2023 6th World Symp. Communication Engineering (WSCE, 2023) 28–33. https://doi.org/10.1109/WSCE59557.2023.10365893

14 Dell PowerEdge: System LED status light indicators on legacy servers (10th and 11th generation): https://www.dell.com (accessed November 2023).

15 A server status indicator LEDs: https://docs.oracle.com (accessed November 2023).

16 Diagnosing the status LEDs – IBM eServer xSeries 240, Netfinity 5600: https://www.ibm.com (accessed November 2023).
17 O. Ronneberger, P. Fischer, and T. Brox: Med. Image Comput Comput.-Assisted Intervention **9351** (2015) 234. https://doi.org/10.1007/978-3-319-24574-4_28
18 K. He, G. Gkioxari, P. Dollar, and R. Girshick: IEEE Trans. Pattern Anal. Mach. Intell. 42 (IEEE, 2020) 386. https://doi.org/10.1109/TPAMI.2018.2844175
19 M. Tarkowski, P. Woznica, and L. Kulas: Proc. 2015 IEEE EUROCON Int. Conf. Computer as a Tool. (IEEE EUROCON, 2015) 1–4. https://doi.org/10.1109/EUROCON.2015.7313723
20 A. S. Abdul Nasir, M. Y. Mashor, and Z. Mohamed: Proc. 2012 IEEE-EMBS Conf. Biomedical Engineering and Sciences. (IEEE-EMBS, 2012) 653–658. https://doi.org/10.1109/iecbes.2012.6498073
21 D. Arthur and S. Vassilvitskii: Proc. 2007 ACM-SIAM Symp. Discrete Algorithms. (ACM, 2007) 1027–1035
22 LabelImg: https://github.com/HumanSignal/labelImg (accessed Nov 2023).

About the Authors

Misheel Enkhbaatar received his B.E. degree from Hacettepe University, Turkey, in 2017 and his M.E. degree from Niigata University, Japan, in 2021. Since 2021, he has been a Ph.D student at Niigata University. His research interests are in image processing, pattern recognition, and machine learning. (f211501d@mail.cc.niigata-u.ac.jp)

Tatsuya Yamazaki received his B.E., M.E., and Ph.D. degrees in information engineering from Niigata University, Japan, in 1987, 1989, and 2002, respectively. From 1989 to 2013, he was a researcher at the National Institute of Information and Communications Technology (formerly Communications Research Laboratory), Japan. Since 2013, he has been a professor at Niigata University. He is also the head of the Big Data Activation Research Center at Niigata University. His research interests are in sensing technologies, image processing, data engineering, machine learning, and pattern recognition. (yamazaki.tatsuya@ie.niigata-u.ac.jp)