1. INTRODUCTION

In recent years, active research has been conducted to enable computers performing creative tasks. In particular, generative models based on deep learning have attracted increasing attention [1, 2]. Applying deep learning has allowed achieving high performance in a wide range of fields, such as speech recognition, image recognition, and natural language processing. The generative models based on deep learning are expected to be used to support creation.

One of the models that can generate images automatically using deep learning is generative adversarial networks (GAN) [3]. GAN is a neural network (NN) that can generate images similar to the learned data. The generator of GAN can be used to generate lifelike images similar to the learned data. The Kansei agent is a model that learns the user’s Kansei evaluation and retrieves images that the user likes. The Kansei agent is constructed using two neural networks. The first network extracts features from given T-shirt images. The second one estimates the user’s evaluation of T-shirt design using the extracted features. We examined the performance of the proposed system according to the results of the evaluation experiment. The results of the experiment showed that the proposed system finally presented most users T-shirt design images that were highly evaluated.

Kansei retrieval is a technique that is used to retrieve the contents that the user wants based on Kansei information. Kansei information includes personal feelings, impressions, and preferences; therefore, it is a complex and fuzzy concept. Kansei retrieval searches the target information after requesting users to select words or photos with sensory expressions. This approach is based on Kansei information corresponding to subjective expression, and therefore, can be useful in the cases when it is difficult to retrieve based on keywords. In the previous studies, Kansei retrieval models have been based on NN to imitate a user’s Kansei [6, 7]. In these models, it is assumed that the input into NN corresponds to the stimulus received by a person, and the output is related to the impression when the person receives the stimulus. Such models are constructed based on a user’s Kansei; they take the information about the search target as an input and then, estimates the user’s evaluation value.

In addition, a Kansei retrieval system has been proposed to extract features using deep learning [8]. The features are extracted from the target data to be handled by the system. When the target data are provided in a form of images, the information about a particular image, such as color and shape, can be extracted as features to distinguish it...
from other images. Extracting features in Kansei retrieval by using deep learning has been confirmed as an efficient approach.

In our previous study, we have proposed a system to generate images matching user’s preference consisting of a Kansei agent and an image generation model based on GAN [9]. The Kansei agent is an NN model that extracts the features of images by using deep learning. Here, the Kansei agent imitates the user’s Kansei and performs Kansei retrieval. In the proposed system of this study, the Kansei agent has evaluated the images generated by the generator trained by GAN. Then, it has retrieved the images matching the user’s Kansei. By using the Kansei agent, we seek to facilitate retrieval according to the user’s preference and to reduce the evaluation burden on users.

In the previous study, an evaluation experiment has been conducted to confirm the effectiveness of the proposed system. In the present paper, we verify the performance of the proposed system according to the results of the evaluation experiment.

2. RELATED WORK

2.1 Generative adversarial networks

Figure 1 represents the flow of data in GAN. GAN is composed of two NNs called as generator and discriminator, which are provided with conflicting learning guidelines [3]. The discriminator is an NN that discriminates between the real images included in the training data and the fake images generated by the generator. The discriminator is aimed to increase the accuracy of discrimination. The generator is an NN that generates the images similar to the specific data corresponding to latent variables. It is aimed to generate the images that the discriminator mistakenly discriminates. During training of GAN, the generator and discriminator are trained alternately and thereby improve the overall performance. By balancing between the learning of the generator and discriminator, it is possible to achieve better final results; namely, the former generates more realistic images, and the latter discriminates more accurately.

Equation (1) describes the objective function of GAN, where $G$ is the generator; $D$ is the discriminator; $x$ denotes the training data; $z$ is a random number called noise or latent variable; $D(x)$ indicates the probability that an input image into the discriminator is derived from the training data; and $G(z)$ indicates the image generated by the generator based on the latent variable. Therefore, $D(G(z))$ is the probability that the generated image is derived from the training data. Consequently, the discriminator aims to increase the objective function, and the generator aims to decrease it.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z(\cdot)} \left[ \log (1 - D(G(z))) \right]$$

(1)

2.2 Convolutional neural network

Convolutional neural network (CNN) is a network in which convolution and pooling processing have been added to NN. Using CNN allows achieving particularly good results in the tasks dealing with the image data [10]. CNN repeats convolution and pooling to capture features of an image. After passing through a fully connected layer by referring to the captured features, the task is finally executed by inputting the features to the activation function. By capturing the input signal in areas, CNN can absorb some changes that do not lose the contents of the image.

Convolution is a process that applies a filter to an image to extract features. A feature map is output by applying a filter to the input image with the set size and interval. The filter scans the entire image, so the features are extracted wherever they are on the image. The size of the feature map is determined by the set interval. In convolution, the weight of the filter changes due to the learning of NN, and the number of output feature maps is determined by the number of filters. Deconvolution is a process in which the signal flow of convolution is reversed.

Pooling is a process that reduces an image while preserving features. Pooling outputs the average or maximum value from the values on the image cut out with the set filter size. The output signal sequence is obtained by scanning this process for the feature map. The size of the output signal sequence is determined by the set interval. In pooling, the operation does not change due to the learning of the NN, and it is applied to one feature map to obtain the output of one image. Pooling also has the effect of increasing robustness of features to the location. The process of expanding, which is the reverse of pooling, is called unpooling or upsampling.
2.3 Kansei retrieval

Kansei retrieval is a technique that is used to retrieve the target based on Kansei information, such as impressions or preferences, without using keywords. Kansei retrieval exploits the subjective Kansei information and is useful in the cases when the search target is difficult to express by objective words.

In the previous studies, Kansei retrieval systems imitating the user’s Kansei and searching the database by using a Kansei agent have been proposed to express the complex Kansei space unique for each person [6, 7]. To learn the user’s Kansei, NN has been used as a part of the Kansei agent. By inputting the stimulus information about user impression of into NN, the Kansei agent can adapt to any type of stimulus and impression. The Kansei agent imitates a human’s Kansei by ensuring that the input unit corresponds to the stimulus information and the output unit corresponds to impression of the stimulus. In this model, the Kansei retrieval has been realized by users through only evaluating the target data. Therefore, it is possible to create a Kansei agent personalized for each user without relying on Kansei words that may have different impressions for each user.

3. PROPOSED SYSTEM

In the present study, we construct an image generation system considering user’s preference. The proposed system consists of an image generation model and a Kansei agent that imitates the user’s Kansei. Figure 2 outlines the structure of the proposed system. The image generation model generates images. Then, the Kansei agent retrieves some generated images and presents them to the user. The Kansei agent estimates the user’s evaluations corresponding to the images generated by the image generation model and presents the selected images. The user evaluates the presented images, and the Kansei agent learns the evaluations accordingly.

The image generation model is trained by using GAN to generate images based on the providing training data. The Kansei agent implemented in the present study is composed of two types of NNs: the one to extract image features, and the other to estimate the evaluations. The users can find the designs they like with less effort and time using the system. Moreover, the system can hold evaluation history and learning for each user.

The Kansei agent is constructed by a feature extraction unit and an evaluation unit. The feature extraction unit has a CNN structure, handles all information of search targets as input signals, and outputs features compressed to the set dimension. The output features are input to the evaluation unit. The evaluation unit receives the input of the features and estimates the user’s evaluation for the image from which the features are extracted. The Kansei agent retrieves the image that the user likes using the estimated value of the evaluation unit. In order for the Kansei agent to imitate the user’s Kansei, it is necessary to optimize the feature extraction unit and the evaluation unit. The initial values of the connection weights of NN are random numbers, and the Kansei agent imitates the Kansei by adjusting the connection weights to appropriate values.

3.1 Image generation

In the present study, we employ progressive growing of GANs (PGGAN), which is an application model of GAN intended for image generation [5]. PGGAN initiates the learning process starting from low resolution and continues it by adding layers so that the resolution is increased gradually. This learning method allows stabilizing the learning and generating high-resolution images. The image generation model incorporated in the proposed system is implemented as a generator trained in advance by using PGGAN.

The trained generator can generate images similar to the given training data using latent vectors of random numbers as input signals. A generated image is derived from the input signal, and the generator can generate an image based on any value in the latent vector space. In addition, when the input signal moves in the latent vector space, it is possible to observe a change in the output image corresponding to the input signal. In other words, the latent vector of the input signal contains the features of the output image. Therefore, the vector space processed by the generator can be considered as the feature space of image features.
3.2 Feature extraction

As a feature extraction unit, CNN is employed to extract features based on RGB values that constitute each pixel of a target input image. Connection weights of the feature extraction unit are optimized by using a convolutional autoencoder (CAE), a model that combines CNN with an autoencoder (AE) [11]. AE is a specific type of NN that learns connection weights so that its input and output signals are the same [12]. AE has a structure in which the number of nodes in the middle layer is reduced so that the output of the middle layer is the encoded representation of the input signal. In other words, AE can be applied to encode an input signal using any number of dimensions. By inputting an expression encoded by AE into the next AE, the encoding procedure is repeated to obtain a high-dimensional expression. As a result, the connection weights required to encode the input signal with any number of dimensions are obtained.

CAE is a special case of the AE model in which convolutional and pooling layers are stacked into multiple layers. Figure 3 represents the example of CAE. In CAE, convolutional and pooling layer are alternately arranged to compress the input signal, and then deconvolutional and unpooling layer are alternately arranged to restore. Since CAE can retain the internal structure of input data, it has high coding performance for image data in which adjacent elements are strongly related. The feature extraction unit optimized by CAE can be used as a network to extract features from images.

3.3 Kansei evaluation

The evaluation unit of the Kansei agent is constructed in a form of NN that estimates the user’s evaluation concerning an image using the features extracted by CNN. This model learns the target user’s Kansei and imitates the corresponding space.

The evaluation unit learns the user’s Kansei by considering the relationship between the features extracted as a result of feature extraction and user’s evaluations concerning input images. The evaluation unit learns to minimize the error between the estimated evaluation, which is the output value, and the user’s evaluation. The data evaluated by a user are added to the evaluation history and stored as the teaching data. The evaluation unit learns all data in the evaluation history and optimizes the connection weights accordingly so as to perform evaluations as the user. If the evaluation unit learns the user’s Kansei successfully, it can estimate the user’s evaluation accurately even concerning the data not evaluated by the user. Therefore, the evaluation unit can estimate the user’s evaluation for all images generated by the generator and then, perform Kansei retrieval based on the estimated evaluations. The evaluation unit can express various Kansei spaces by learning the connection weights corresponding to the appropriate values. This model can be used to build an individual model for each user and to learn the corresponding user’s Kansei.

4. EXPERIMENT

4.1 Experimental setup

In the present study, we verified the effectiveness of the proposed system by conducting an experiment involving real users. The subjects of this experiment were 13 men and women in their 20s. We tested whether the proposed system could learn the user’s Kansei and present the T-shirt images that match the user’s preference. First, we trained the image generation model and feature extraction unit using the prepared T-shirt image dataset. Second, we optimized the evaluation unit using real user’s evaluations and verified the model performance.

In this experiment, the subjects evaluated T-shirt images generated by the image generation model trained by PGGAN. The subjects provided their evaluations according to a ten-point scale based on “whether or not they like.” The number of images to be evaluated by a subject was ten per one generation. The evaluated data were stored in the evaluation history as the teaching data. Thereafter, the Kansei agent learned all data stored in the evaluation history, including the newly stored data. After learning the subject’s Kansei, the Kansei agent evaluated all T-shirt images generated by the image generation model, and then, presented the selected images to the subject based on the evaluations. In this experiment, the image generation model generated 5,000 T-shirt images. In addition, the subject’s evaluations did not affect the generation process of PGGAN, and the distribution of the generated images did not change.

After the subject completed the evaluation for ten generations, the learning process of the evaluation unit was finished, and the final images were presented. The subject also evaluated the final presented images on the ten-point scale to verify whether the proposed system presented the T-shirt images aligned to the subject’s preference. The final presented images were the five
T-shirt design variants estimated by the Kansei agent as the most favorable ones for the subject.

In this experiment, we collected T-shirt images mainly using Web image search and included 1,130 images of the same size and shape into the image dataset. Figure 4 represents the examples of T-shirt images used as the training data. The images with the size of 128 × 128 pixels were used to train the image generation model and feature extraction unit. The subject’s evaluations for the presented T-shirt images were used to train the evaluation unit of each subject.

Figure 5 represents the interface implemented in this experiment. Here, the system presented a total of ten images: five were selected based on the evaluation by the Kansei agent and other five were randomly selected to increase diversity. Each subject evaluated the ten displayed T-shirt images based on the personal preference. As the images displayed in the first generation were randomly selected, the T-shirt design variants to be evaluated were different for each subject. After the subject completed the evaluation for the ten generations, the final selected images were displayed at the bottom of the window and then, evaluated by the subject.

4.2 Experimental conditions

The image generation model was trained in advance by PGGAN. Table 1 outlines the learning conditions set for PGGAN. The training data were 1,130 T-shirt images with the size of 128 × 128 pixels. In this experiment, the total number of learning was 1,000,000. This is equivalent to about 900 epochs if we consider by the number of learning epochs. The activation function of all layers except the output layer that performed linear transformation was set as leaky rectified linear unit (LReLU) function. The model was optimized by using Adam, an applied method of stochastic gradient descent [13]. Adam’s parameters were set as follows: $\alpha=0.001$; $\beta_1=0$; $\beta_2=0.99$; $\varepsilon=10^{-8}$. The batch size was changed as the learning progressed. The batch size was 128 up to the resolution of 16 × 16, and thereafter, it was halved every time the resolution increased, and the batch size became 16 at the resolution of 128 × 128.

Figure 6 represents the final generator structure of PGGAN used in this experiment. The generator received 512-dimensional latent variable as an input signal, and repeated the two convolutional and upsampling layers.

| Table 1: Learning of PGGAN |
|----------------------------|
| Training data          | 1,130 |
| Weight initialization  | Random|
| Activation function    | LReLU |
| Optimization method    | Adam  |
| Total number of learning | 1,000,000 |
| Batch size             | 16–128 |
| Loss function          | Cross entropy |
to output an image having the size of 128×128 pixels. The structure of the discriminator was the inverse of that of the generator. The discriminator compressed the information by repeating two convolutional and pooling layers concerning the inputted image. The compressed input signal was converted to 512 dimensions and connected to the fully connected layer to identify the input image.

The feature extraction unit was trained in advance by CAE. Table 2 outlines the learning conditions set for CAE. It was trained using 1,130 prepared training data. The activation function of layers other than the output layer was defined as the ReLU functions, and that of the output layer was the sigmoid function. The model was optimized by using Adam in which the parameters were set according to the values recommended in the paper where it was proposed originally. In addition, batch normalization was applied to all convolutional layers to equalize the signal distribution of each minibatch [14].

Figure 7 represents the structure of the feature extraction unit. In the present study, the proposed system considered RGB pixel values in a 128×128 pixels image as an input signal of the feature extraction unit; therefore, the input signal had 128×128×3 dimensions. In the middle layer, the input signal was encoded to a lower dimension each time the signal progressed to the next layer. Finally, the output layer outputted the features compressed to 16 dimensions.

The evaluation unit was optimized using the subject’s evaluations and the features extracted by the trained feature extraction unit. Table 3 outlines the learning conditions set for the evaluation unit. The evaluation unit learned for ten epochs each time the subject evaluated one generation of images. The training data used for the evaluation unit were the images generated by the generator and presented to the subject; therefore, ten images were added each time the subject evaluated the images in one generation. The activation function of layers other than the output layer was the ReLU function, and that of the output layer was the sigmoid function. The model was optimized by using Adam in which the parameters were set according to the values recommended in the paper where it was proposed originally.

Figure 8 represents the structure of the evaluation unit. All layers of the evaluation unit were fully connected layers. The number of nodes in the output layer was equal to one, and the output value was between 0.0–1.0, which corresponded to the range of the sigmoid function. As the user’s evaluation value was on the ten-point scale of 1–10, the output value range of the evaluation unit was divided into ten levels, and the median of the corresponding output value sections was used as the teaching signal.

4.3 Results and consideration

Table 4 shows the subjects’ evaluations for the final presented images, the average evaluation errors between each subject and Kansei agent for the presented images, and the variances of each subject’s evaluations in ten

| Table 3: Learning of the evaluation unit |
|-----------------------------------------|
| Training data | 100 |
| Weight initialization | Random |
| Activation function | Output layer: Sigmoid, Other: ReLU |
| Optimization method | Adam |
| Learning epochs | 100 |
| Batch size | 10 |
| Loss function | Mean square error |

![Figure 7: Structure of the feature extraction unit](image)

![Figure 8: Structure of the evaluation unit](image)
generations. Many subjects highly evaluated the presented images; therefore, it is considered that the Kansei agent presented images matching the subjects’ preferences. Furthermore, many subjects had small evaluation errors, and most of the subject’s average evaluation errors are less than two. Figure 9 represents the numbers of images for each error value between the subject’s evaluation and the learned Kansei agent’s estimated one. The numbers of images provided are the averages of all subjects. As shown in Fig. 9, the error is two or less for more than 70% of the presented images. In Kansei retrieval, the user’s evaluations are provided according to a rough sense, and therefore, evaluation fluctuations may occur. Considering the users’ evaluation fluctuations, the errors of two or less are considered to be rather insignificant.

As shown in Tab. 4, the subjects who provided 10 points for the final presented image tend to have large variances of evaluations. In addition, the subjects who had large variances of evaluations tend to have small evaluation errors. It is considered that the subjects with large variances of evaluations have clearly identifiable preferences. The Kansei agent learned better about Kansei of the subjects who had clear preferences than those that had unclear preferences.

On the other hand, as shown in Tab. 4, some subjects did not highly evaluate the final presented images and had large evaluation errors. In other words, for these subjects, the Kansei agent did not learn the subject’s Kansei well. It is considered that the images with effective features for these subjects did not appear because the spaces that needed to be searched were too large. Therefore, it is necessary to develop a model that has high effectiveness for all users.

Figure 10 represents the learning result of the evaluation unit. Here, the horizontal axis shows the number of learning epochs, and the vertical axis shows the mean square error in the learning of the evaluation unit. The values provided in the figure are the averages of all subjects and are plotted at each learning epoch. As the output value range of the evaluation unit is 0.0–1.0, the value range of the loss function is also 0.0–1.0. The mean square error corresponds to error evaluations in the learning of the evaluation unit. A decrease in the error value indicates that the evaluation unit has improved the accuracy of estimating the user’s evaluation. As shown in Fig. 10, the error decreases and converges as the learning progresses. Therefore, it is concluded that the Kansei agent has learned the subjects’ Kansei.

Figure 11 represents the examples of the highly evaluated images, low evaluated images, and final presented images.
images selected among those presented to one of the subjects during this experiment. The numbers below the images denote the subject’s evaluations for each image. The subject provided high evaluations for the designs that had pictures or patterns on black and white bases. The Kansei agent finally presented the images with similar characteristics, and the subject highly evaluated those images. Some of the final presented images were evaluated as low by the subjects; however, the Kansei agent presented most of subjects the images that high evaluations were provided. It is considered that the Kansei agent learned the subject’s preferences to a particular extent.

We compared the evaluations for the images presented by the Kansei agent and those presented randomly to examine the effectiveness of the proposed system. The images presented in the first generation in the experiment were randomly selected, so we compared the evaluations for the first and the final presented images. Table 5 shows the averages of each subject’s evaluations. Most subjects’ evaluations increased from the first generation. In addition, Wilcoxon signed-rank test was used on the evaluations of first generation (Random) and final images (Agent) shown in Table 5 [15]. This result revealed a significant difference between the evaluations for the images presented by the agent and those presented randomly ($p=1.7\times10^{-3}$).

| Subject | Random | Agent |
|---------|--------|-------|
| A       | 3.6    | 6.2   |
| B       | 3.7    | 6.0   |
| C       | 4.6    | 6.2   |
| D       | 4.4    | 5.2   |
| E       | 4.3    | 7.4   |
| F       | 4.3    | 4.8   |
| G       | 4.4    | 6.4   |
| H       | 5.7    | 8.6   |
| I       | 3.9    | 8.2   |
| J       | 6.1    | 4.8   |
| K       | 3.8    | 8.2   |
| L       | 4.9    | 5.6   |
| M       | 5.4    | 7.8   |

| Subject | Random | Agent |
|---------|--------|-------|
| A       | 4.6    | 6.2   |
| B       | 4.4    | 5.6   |
| C       | 4.6    | 6.2   |
| D       | 4.4    | 5.2   |
| E       | 4.3    | 7.4   |
| F       | 4.3    | 4.8   |
| G       | 4.4    | 6.4   |
| H       | 5.7    | 8.6   |
| I       | 3.9    | 8.2   |
| J       | 6.1    | 4.8   |
| K       | 3.8    | 8.2   |
| L       | 4.9    | 5.6   |
| M       | 5.4    | 7.8   |

Table 5: Comparison of evaluation average

5. CONCLUSION

In the present paper, analyzing the results of the conducted evaluation experiment, we examined the performance of the image generation system based on GAN and a Kansei agent. It was confirmed that the Kansei agent learned the user’s Kansei for the generated images with an acceptable level of performance. In addition, it was confirmed that the proposed system finally presented the T-shirt design images that were highly evaluated to most users. Furthermore, the effectiveness of the proposed system was confirmed by the test. In the future, it is expected that a more practical system will be constructed through the development of the Kansei agent model facilitating more efficient learning and high-precision estimation, and improvement of the generation model to diversify generated images and improve the image quality.

REFERENCES

1. Elgammal, A., Liu, B., Elhoseiny, M., and Mazzone, M.; CAN: Creative Adversarial Networks, Generating “Art” by learning about styles and deviating from style norms, arXiv:1706.07068, 2017.
2. Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S., and Eck, D.; Onsets and frames: Dual objective piano transcription, arXiv:1710.11153, 2017.
3. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.; Generative adversarial nets, Advances in Neural Information Processing Systems 27, pp.2672-2680, 2014.
4. Radford, A., Metz, L., and Chintala, S.; Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv:1511.06434, 2015.
5. Karras, T., Aila, T., Laine, S., and Lehtinen, J.; Progressive growing of GANs for improved quality, stability, and variation, arXiv:1710.10196, 2017.
6. Urai, T., and Tokumaru, M.; User Kansei clothing image retrieval system, Journal of Advanced Computational Intelligence and Intelligent Informatics, 18(6), pp.1044-1052, 2014.
7. Inoue, M., Takenouchi, H., and Tokumaru, M.; Music recommendation system improvement using distributed genetic algorithm, 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 17th International Symposium on Advanced Intelligent Systems, pp.627-630, 2016.
8. Ota, S., Takenouchi, H., and Tokumaru, M.; Kansei clothing retrieval system using features extracted by autoencoder, 2017 IEEE Symposium Series on Computational Intelligence Proceedings, pp.281-287, 2017.
9. Kaji, N., Takenouchi, H., and Tokumaru, M.; T-Shirt Image generation system considering user preference using generative adversarial networks, The 20th
International Symposium on Advanced Intelligent Systems and 2019 International Conference on Biometrics and Kansei Engineering, pp.158-165, 2019.

10. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A.; Going deeper with convolutions, 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp.1-9, 2015.

11. Masci, J., Meier, U., Ciresan, D., and Schmidhuber, J.; Stacked convolutional auto-encoders for hierarchical feature extraction, International Conference on Artificial Neural Networks, pp.52-59, 2011.

12. Hinton, G.E., and Salakhutdinov, R.R.; Reducing the dimensionality of data with neural networks, Science, 313(5786), pp.504-507, 2006.

13. Kingma, D.P., and Ba, J.; Adam: A method for stochastic optimization, arXiv:1412.6980, 2014.

14. Ioffe, S., and Szegedy, C.; Batch normalization: Accelerating deep network training by reducing internal covariate shift, Proceedings of the 32nd International Conference on International Conference on Machine Learning, 37, pp.448-456, 2015.

15. Wilcoxon, F.; Individual Comparisons by ranking methods, Biometrics Bulletin, 1(6), pp.80-83, 1945.

Naoki KAJI (Non-member)
Naoki Kaji received the B.E. and M.E. degrees in Engineering from Kansai University in Osaka, Japan in 2018 and 2020, respectively. His research interests include generative model and Kansei retrieval.

Hiroshi TAKENOUCHI (Member)
Hiroshi Takenouchi is an Assistant Professor of Faculty of Information Engineering at Fukuoka Institute of Technology, Japan where he has worked since 2014. He received B.E., M.E. and D.E. degrees in Engineering from Kansai University in 2008, 2010 and 2013, respectively. In 2013, he had been a part-time researcher and postdoctoral fellow of Kansai University. His current research interests include fuzzy reasoning and an application with Interactive Evolutionary Computation. He is a member of Japan Society for Fuzzy Theory and Intelligent Informatics, Japan Society of Kansei Engineering, Institute of Electronics, Information and Communication Engineers and IEEE.

Masataka TOKUMARU (Member)
Masataka Tokumaru is a Professor of Faculty of Engineering Science at Kansai University in Osaka, Japan since 2016. He received the B.E. and M.E. degrees in Engineering from Kansai University in 1994 and 1997. During 1997-2007, he had been a Research Associate or a Lecturer of Department of Electronics at Kansai University. He received Dr. (Eng.) degree from Osaka City University, Japan in 2001. His research interests are affective computing and evolutionary computation. He is a member of Japan Society of Kansei Engineering, Institute of Electronics, Information and Communication Engineers, Japan Society for Fuzzy Theory and Intelligent Informatics, Human Interface Society, Japanese Society for Artificial Intelligence and IEEE.