Evaluation of Transfer Learning for Handwritten Character Classification Using Small Training Samples

Yoshihiro Mitani [1,*], Naoki Yamaguchi [1], Yusuke Fujita [2], and Yoshihiko Hamamoto [2]

[1] National Institute of Technology, Ube College, Ube, Japan
[2] Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube, Japan
*Correspondence: mitani@ube-k.ac.jp (Y.M.)

***Abstract*—In pattern recognition fields, it is worthwhile to develop a pattern recognition system that hears one and knows ten. Recently, classification of printed characters that are the same fonts is almost possible, but classification of handwritten characters is still difficult. On the other hand, there are a large number of writing systems in the world, and there is a need for efficient character classification even with a small sample. Deep learning is one of the most effective approaches for image recognition. Despite this, deep learning causes overtrains easily, particularly when the number of training samples is small. For this reason, deep learning requires a large number of training samples. However, in a practical pattern recognition problem, the number of training samples is usually limited. One method for overcoming this situation is the use of transfer learning, which is pretrained by many samples. In this study, we evaluate the generalization performance of transfer learning for handwritten character classification using a small training sample size. We explore transfer learning using a fine-tuning to fit a small training sample. The experimental results show that transfer learning was more effective for handwritten character classification than convolution neural networks. Transfer learning is expected to be one method that can be used to design a pattern recognition system that works effectively even with a small sample.**

***Keywords*—small training sample sizes, handwritten character classification, transfer learning, fine-tuning, convolution neural networks**

I. Introduction

A considerable amount of effort has been devoted to designing classifiers that use small training sample sizes [1−3]. It is known that the larger the training samples, the better the classification performance of a classifier. However, the sizes of the available samples are usually limited, and it is difficult to correct many samples in order properly to train a classifier, as this is a time-consuming task. Therefore, a label or class name must be given for each sample. Because of this, it is desirable to develop a pattern recognition system that can work well even for small training samples. In this study, a situation is defined to have a small training sample size if the ratio of the number of training samples to the dimensionality is less than 1. The larger the ratio, the greater the number of training samples, and the more convenient it is to develop a pattern classification system. Conversely, the smaller the ratio, the fewer the number of training samples, and the more difficult it is to develop a pattern classification system. In particular, when the ratio is less than 1, the inverse of the sample covariance matrix is impossible to compute, making it difficult to design a classifier such as, a Linear Discriminant Analysis (LDA) [2]. The ratio must be larger than five before the bias in the design-set error rate is sufficiently small [1].

Convolutional Neural Networks (CNNs) originate from artificial neural networks [4]. CNNs have been successfully applied in the field of image recognition [5, 6]. CNNs have been reported as having promising classification performance for handwritten digit classification problems [7, 8]. One advantage of CNNs is that they can automatically learn a relationship between input and output, which eliminates the need to know what features or classifier should be used. Instead of properly training many hyperparameters in a CNN, many training samples are required. There is a trade-off between generalization performance and overtraining. In a practical pattern recognition problem, the number of available samples is usually limited. Therefore, it is important to design a CNN to adapt to situations with small training sample sizes. Keshari *et al.* [9] have addressed this issue by focusing on learning the structure and strength of filters.

Transfer learning, the deep net of which is pretrained by many samples, can be applied to other pattern recognition problems, particularly when the number of training samples is small. Thus far, various types of transfer learning have been developed [10, 11].

Handwritten character classification is one application of transfer learning. However, it is difficult to develop a handwritten character classification system because handwritten characters are not easily available and the number of samples is generally small. In particular, it is more difficult if the handwritten characters are highly distorted. Additionally, it is not clear which transfer

Manuscript received June 6, 2022; revised July 21, 2022; accepted August 11, 2022.

doi: 10.18178/joig.11.1.21-25

learning method is effective for handwritten character classification. In this study, we present the evaluation of transfer learning for handwritten character classification of Mixed National Institute of Standards and Technology Database (MNIST) [12] and Kuzushiji-MNIST (KMNIST) [13] datasets using a training sample with a small size. An example of research on KMNIST is the development of a U-Nets based model that predicts character position and character type from classical images using KuroNet [14]. There are also studies on generative adversarial networks (GANs) that generate new character types or styles [15−17]. The experimental results show that transfer learning is more effective for handwritten character classification with a small training sample size than CNNs. Our result suggests using transfer learning is helpful in pattern recognition problems with small training samples.

II. Methods

A. Handwritten Characters

In this study, we used the MNIST [12] and KMNIST (Kuzushiji-MNIST) image datasets [13]. The MNIST dataset consists of images of handwritten digit characters. The KMNIST dataset consists of handwritten classical Japanese character images and was developed by referring to the MNIST dataset. Fig. 1 shows examples of images in KMNIST. Compared to the handwritten digits in the MNIST dataset, the handwritten Kuzushijis in the KMNIST dataset have been heavily distorted because they were written by hand over 150 years ago. In general, even Japanese people cannot read the letters easily. This means the classification problem for the KMNIST dataset is more difficult than that of the MNIST. For each dataset, we used at most 20000 images. This was a 10-class problem. We assumed there were no imbalanced classes. Thus, we used 1000 test images and at most 1000 training images for each class. We used 100 training images for each class to ensure the training sample sizes were small. The image size was 28×28. Thus, the ratio of the training sample size to the dimensionality was 0.1276 (100/784). The number 784 was the dimensionality of the image size (28×28). Because the ratio was less than 1, this situation was considered to have a small training sample size. The images were in gray scale. Before feeding the training images to the deep net, the image size was changed from 28×28 to 150×150 in order to fit the input requirement of the deep net.

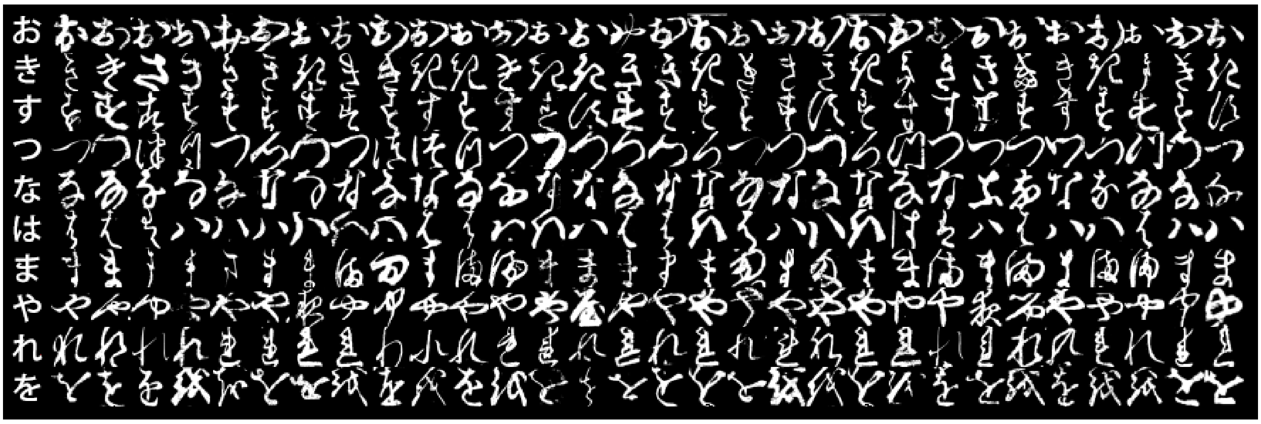

Figure 1. Examples of KMNIST images.

B. CNN

We evaluated the CNN in terms of its generalization performance for the handwritten digit characters from MNIST and the classical Japanese characters from KMNIST. The generalization performance of a CNN depends on its network structure and the parameters to be determined. Referring to the CNN [18], we determined that our CNN network structure was almost the same as the original CNN. Fig. 2 shows a network structure of the CNN we used. The initial sizes of the MNIST and the KMNIST images were 28 times 28. In the experiment, we changed the input image size to 150×150. First, we convolved the image by using 32 filters with a 3×3 filter size. The image size was then 148×148. Using 2×2 maxpooling, we reduced the image size by half to 74×74. Second, we convolved the image by using 64 filters with a 3×3 filter, and performed maxpooling in the same manner. The image size changed forms 72×72 to 36×36. Third, we again convolved the image by using 64 filters with a 3×3 filter. The image size was then 34×34. Fourth, we flattened this image was then into 73984 dimensional (64×34×34) data. Finally, we constructed a fully connected artificial neural network. The network had one hidden layer. The number of the neurons also depended on the generalization performance of the CNN. In the experiment, we used 64 for simplicity. Then we used dropout, the rate of which was 0.25. The number of the outputs of the CNN was 10, because this was a 10-class problem, as mentioned above. Therefore, the structure of the fully connected artificial neural network was 73984-64-10. All the activation functions were ReLU except for the output. In the output, we used softmax, and the learning optimizer was Adam. The epochs and batch size were 50 and 32, respectively.

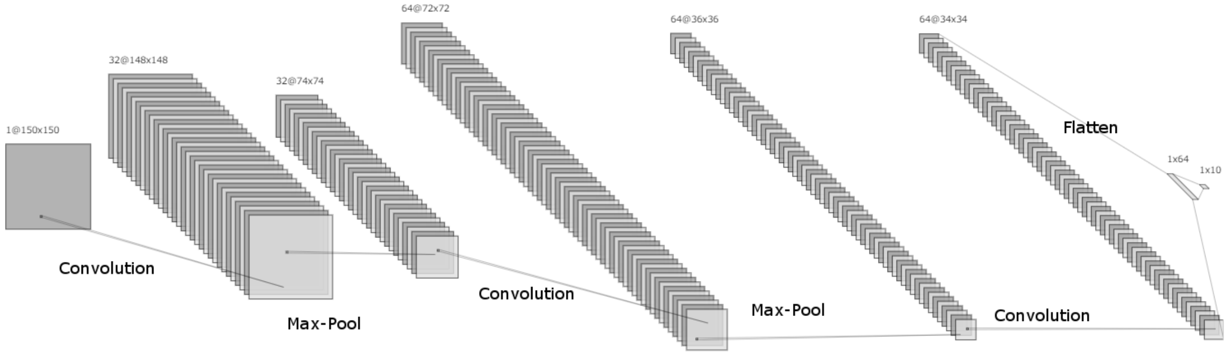

Figure 2. CNN structure used in this study.

C. Transfer Learning

In general, transfer learning performs well for pattern recognition problems with small sample sizes. Therefore, we expected transfer learning to perform well when classifying handwritten digit and classical Japanese characters using a small training sample. Thus far, we have never seen a specific classifier outperforms others for every pattern recognition problem. Therefore, we investigated a variety of transfer learning methods. In this study, we used the following methods: VGG16, VGG19, InceptionV3, DenseNet121, DenseNet169, DenseNet201, and InceptionResNetV2 [11]. These were already pre-trained by ImageNet [19] with large samples. In order to fit each of the MNIST and KMNIST classification problems to each network of the transfer learning methods, the fine-tuning technique was used. In general, the generalization performance of transfer learning depends on how the networks are tuned by the given training samples. Therefore, we conducted five types of fine-tuning, as described below. The fine-tuning techniques were denoted by levels 0, −1, −2, −3, and −4, for each network of transfer learning. For each deep net, each level was divided into two groups: freeze and tuned networks. Table I summarizes the numbers of overall and freeze networks for each fine-tuning level (“overall” means the total number of freeze and tuned networks). Level 0 represents the situation in which only the fully connected artificial neural network was tuned by our training samples. The smaller the value of the level, the larger the number of fine-tuned networks. The percentage of the tuned networks for level −1 was larger than that of level 0, and that of level −2 was larger still. Therefore, the effect of each level of fine-tuning in transfer learning needs to be investigated.

TABLE I. NUMBERS OF OVERALL AND FREEZE NETWORKS FOR EACH FINE-TUNING LEVEL OF TRANSFER LEARNING

	Level	0	−1	−2	−3	−4
	overall	freeze	freeze	freeze	freeze	freeze
VGG16	23	19	15	11	7	4
VGG19	26	22	17	12	7	4
I.V3	315	311	280	249	229	197
D.N.121	431	427	313	141	53	7
D.N.169	599	595	369	141	53	7
D.N.201	711	707	481	141	53	7
I.R.N.V2	784	780	618	275	41	1

III. RESULTS AND DISCUSSION

We used at most 20000 available MNIST [12] and KMNIST [13] images. Both the dimensionalities are 784 because the image size was 28 by 28. In the experiments, we used at most 1000 training images and 1000 test images per class for each dataset. We used 100 training images (a small sample size) for each class. As mentioned above, we assumed a situation in which the ratio of the number of training samples to the number of dimensions was smaller than 1. When we used 100 training images per class, the ratio was 0.1276 (100/784), which is less than 1.

We evaluated the generalization performance of the deep nets according to the error rate. The error rate was defined as the ratio of the number of misclassified test images to the total number of test images. For error rate estimation, the holdout method has been successful, because it maintains the statistical independence between the training and test images [20, 21]. To evaluate the generalization performance of the deep nets, the average error rate was obtained by using the holdout method. Fig. 3 shows the flow of the error rate estimation computed using the holdout method. First, we randomly divided at most 20,000 available images into at most 10,000 training images and 10000 test images. Second, we trained the classifier with at most 10,000 training images. Next, using the classifier, we computed the error rate with 10,000 test images. By repeating these steps five times, we obtained the average error rate and the standard deviation.

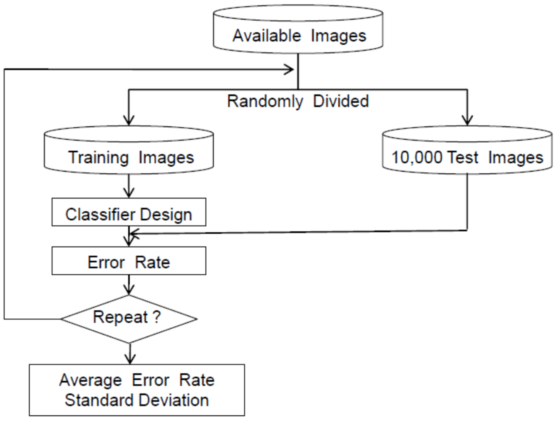

Figure 3. Flow of the error rate estimation using the holdout method.

Next, we found a baseline of the generalization performance of the CNN for the MNIST and the KMNIST datasets in terms of the average error rate. For each class, we varied the training image sizes to 100, 200, 500, and 1000. The ratios of the training sample sizes to the dimensionality were 0.1276 (100/784), 0.2551 (200/784), 0.6376 (500/784), and 1.2755 (1000/784). We decided to fix the test image size as 1000 per class. Table II shows the average error rate of the CNN as a function of the number of training images per class. As expected, the larger the number of training images, the better the generalization performance of the CNN. The best average error rates were 3.11% for the MNIST dataset and 6.35% for the KMNIST dataset, both of which occurred when the number of training images per class was 1000. In contrast, the poorest average error rates were 9.82% for the MNIST and 19.84% for the KMNIST, both of which occurred when the training image size was 100 for each class. In the following experiments, we used 100 training images per class. We assumed this was a situation with a small training sample size. The distortion of the KMNIST was larger than that of the MNIST, not only by the appearance of images but also in terms of the average error rate.

TABLE II. AVERAGE ERROR RATE OF THE CNN AS A FUNCTION OF THE NUMBER OF TRAINING IMAGES PER CLASS

Upper rows: average error rate (%), lower rows: standard deviation

(a) MNIST dataset				
	100	200	500	1000
MNIST	9.82 0.84	6.63 0.49	4.41 0.31	3.11 0.18
(b) KMNIST dataset				
	100	200	500	1000
KMNIST	19.84 0.26	13.60 0.27	9.26 0.35	6.35 0.19

Then, we examined the generalization performance of transfer learning using a small training sample size. We compared the following transfer learning methods to the baseline of the generalization performance of the CNN: VGG16, VGG19, InceptionV3, DenseNet121, DenseNet169, DenseNet201, and InceptionResNetV2 [11]. Table III shows the average error rates of the deep nets for each fine-tuning level when the number of training images was 100 per class. For both the MNIST and KMNIST datasets, as the level of fine tuning was reduced (i.e., as the number of freeze networks was reduced), the average error rate decreased or was saturated. Therefore, in order to design transfer learning, the fine-tuning of the deep nets should be considered. As shown in Table III(a), we found the minimum average error rate for the MNIST data to be 2.67% when we used VGG16 with a fine-tuning level of −3. This deep net exhibited the best performance among our limited experiments with the MNIST dataset. The generalization performance of VGG16 with an average error rate of 2.67% dramatically outperformed the CNN with an error rate of 9.82%. In contrast, as shown in Table III(b), the minimum average error rate for the KMNIST data was 5.20% when we used InceptionResNetV2 with a fine-tuning level of −4. This transfer learning method demonstrated the best performance among our limited experiments with the KMNIST dataset. The generalization performance of InceptionResNetV2 with an average error rate 5.20% was significantly higher than that of the CNN with an average error rate of 19.84%.

TABLE III. AVERAGE ERROR RATES OF DEEP NETS FOR EACH FINE-TUNING LEVEL WHEN THE NUMBER OF TRAINING IMAGES WAS 100 PER CLASS

Upper rows: average error rate (%), lower rows: standard deviation

(a) MNIST dataset					
Level	0	−1	−2	−3	−4
VGG16	9.04 0.31	3.73 0.22	3.01 0.24	2.67 0.28	2.83 0.32
VGG19	8.95 0.29	3.35 0.18	2.69 0.24	2.75 0.28	2.80 0.52
I.V3	26.46 8.84	15.55 5.60	9.45 0.67	9.02 0.17	7.61 3.69
D.N.121	5.84 0.17	5.29 0.48	3.59 0.46	3.23 0.40	3.15 0.21
D.N.169	5.55 0.21	4.93 0.40	3.52 0.19	3.51 0.36	3.22 0.38
D.N.201	5.22 0.17	5.12 0.28	3.31 0.38	3.42 0.36	3.24 0.27
I.R.N.V2	10.16 0.96	5.74 0.17	3.23 0.37	2.75 0.18	2.68 0.41
(b) KMNIST dataset					
Level	0	−1	−2	−3	−4
VGG16	24.42 0.73	7.95 0.44	7.22 0.49	6.19 0.58	6.32 0.74
VGG19	25.22 0.80	7.24 0.26	6.50 0.59	6.11 0.49	7.66 2.46
I.V3	37.36 1.53	25.12 1.01	17.99 0.37	16.76 0.29	13.52 1.44
D.N.121	14.67 0.34	11.76 0.77	8.54 0.35	7.78 0.48	7.78 0.63
D.N.169	13.90 1.12	10.26 0.48	8.41 1.11	7.35 0.20	7.43 0.34
D.N.201	12.32 0.51	10.87 0.58	7.49 0.29	7.31 0.43	7.19 0.68
I.R.N.V2	23.65 1.87	11.62 0.99	5.62 0.52	5.35 0.40	5.20 0.44

According to the results displayed in Tables II and III, the generalization performances of the VGG16 with a fine-tuning level of −3 and InceptionResNetV2 with a fine-tuning level of −4, both of which used small training samples sizes (100), were superior to that of the CNN with large training samples (1000). Therefore, in order to solve small training sample size problems, such as the one described in this study (i.e., the classification of handwritten digit and classical Japanese characters), the transfer learning approach should be used. The fine-tuning of transfer learning should also be taken into account. We found that there are certain models of transfer learning that are more effective for different datasets. Therefore, it is important to carefully select the appropriate transfer learning model and fine-tune it to suit the data.

IV. CONCLUSION

In this paper, we presented the evaluation of transfer learning for classifying handwritten digit characters in the

MNIST dataset and handwritten classical Japanese characters in the KMNIST dataset using a small training sample. We defined situations with small training sample sizes as those with a ratio of the number of training samples to the dimensionality of less than 1. The experimental results showed that the transfer learning classified both the handwritten digit and classical Japanese characters more effectively than the CNNs. Thus, in order to design an effective handwritten character classification system, we recommend using transfer learning with fine-tuning. Transfer learning is expected to be one method that can be used to design a pattern recognition system that works effectively even with a small sample. Our study suggests that transfer learning is one highly effective method that can be used for practical pattern recognition problems.

In future work, we will investigate applying transfer learning to another type of pattern recognition problem, such as recognizing Kanji characters [13]. Images of handwritten Kanji characters have feature much more distortion and have many classes that require classification. This may be a more challenging task, particularly with a small training sample. Furthermore, we will explore another type of image recognition problem with small samples to investigate the performance of transfer learning when applied to other images.

Conflict of Interest

The authors declare no conflict of interest.

Author contribution

All authors conducted the research. Y. Mitani wrote the basic sentences of the paper. N. Yamaguchi executed the experiments. Y. Mitani created the images. Y. Mitani, Y. Fujita, and Y. Hamamoto analyzed the data. All authors concluded the results. Y. Mitani supervised N. Yamaguchi. All authors had approved the final version.

References

[1] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, P. R. Krishnaiah and L. N. Kanal, Eds., North-Holland Publishing Company, 1982, vol. 2, pp. 835–855.

[2] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners and open problem," in *Proc. 10th International Conference on Pattern Recognition*, Atlantic City, 1990, pp. 417–423.

[3] S. J. Raudys and A. K. Jain, "Small sample size problems in designing artificial neural networks," *Machine Intelligence and Pattern Recognition*, vol. 11, pp. 33-50, 1991.

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back propagation errors," *Nature*, vol. 323, no. 9, pp. 533–536, 1986.

[5] G. E. Hinton, S. Osindero, and Y. W. The, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 28, pp. 436–444, 2015.

[7] K. Kumar and H. Beniwal, "Survey on handwritten digit recognition using machine learning," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 5, pp. 96–100, 2018.

[8] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," arXiv:1905.03288, 2019.

[9] R. Keshari, M. Vatsa, R. Singh, *et al.*, "Learning structure and strength of CNN filters for small sample size training," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 2018, pp. 9349–9358.

[10] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A comprehensive survey on transfer learning," arXiv:1911.02685, 2019.

[11] Keras Applications. (2022). [Online]. Available: https://keras.io/api/applications

[12] Y. LeCun, L. Bottou, Y. Bengio, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, *et al.*, "Deep learning for classical Japanese literature," arXiv:1812.01718, 2018.

[14] A. Lamb, T. Clanuwat, and A. Kitamoto, "KuroNet: Regularized residual U-nets for end-to-end Kuzushiji," *SN Computer Science*, vol. 1, no. 3, 2020.

[15] J. Zeng, Q. Chen, Y. Liu, *et al.*, "StrokeGAN: Reducing mode collapse in Chinese font generation via stroke encoding," arXiv.2012.08687, 2022.

[16] Y. Zhang, Y. Zhang, W. Cai, *et al.*, "Separating style and content for generalized style transfer," arXiv:1711.06454, 2018.

[17] C. Li, Y. Taniguchi, M. Lu, *et al.*, "Few-shot font style transfer between different languages," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 433–442.

[18] Google Brain. (2022). TensorFlow. [Online]. Available: https://www.tensorflow.org

[19] ImageNet. (2022). [Online]. Available: https://image-net.org/

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., New York, USA: Wiley Interscience, 2001.

[21] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., San Diego, USA: Academic Press, 1990.

Yoshihiro Mitani received the B. S., M. S., and Ph D. degrees from Yamaguchi University, Japan, in 1993, 1995, and 1999, respectively. He is currently a professor at the National Institute of Technology, Ube College, Ube, Japan. His research interests include pattern recognition and image processing techniques. He is a member of IEEE.

Naoki Yamaguchi received the associate degree from National Institute of Technology, Ube College, Ube, Japan, in 2021. He is currently an advanced course student at the National Institute of Technology, Ube, College, Ube, Japan. His research interests include image processing techniques and machine learning.

Yusuke Fujita received the B.E., M.E. and Ph.D. degrees from Yamaguchi University, Japan, in 2003, 2005 and 2008, respectively. He is currently an associate professor at Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube, Japan. His research interests include computer vision, image processing, pattern recognition, and applied deep learning. He is a recipient of the 2011 JSCE Paper Encouragement Award from Japan Society of Civil Engineers. He is a member of IEEE.

Yoshihiko Hamamoto received the B.S., M.S. degrees from Yamaguchi University, Japan, in 1981, 1983, respectively. He received the Ph.D. degree from the Tokyo Institute of Technology in 1992. In 1983 he joined NEC Corporation, where he worked on the development of an optical character reader. He is currently a professor at Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Ube, Japan. His research interests are in statistical pattern recognition.