Received: 30 January 2018 | Revised: 30 April 2018 | Accepted: 14 May 2018

DOI: 10.4218/etrij.2018-0066

SPECIAL ISSUE

WILEY ETRI Journal

Human-like sign-language learning method using deep learning

Yangho Ji[1] | Sunmok Kim[1] | Young-Joo Kim[2] | Ki-Baek Lee[1]

[1]Department of Electrical Engineering, Kwangwoon University, Seoul, Rep. of Korea.

[2]Logistics System Research Team, Korea Railroad Research Institute, Seoul, Rep. of Korea.

Correspondence
Ki-Baek Lee, Department of Electrical Engineering, Kwangwoon University, Seoul, Rep. of Korea.
Email: kblee@kw.ac.kr

Funding information
Research Grant of Kwangwoon University 2016; Human Resources Program in Energy Technology, Ministry of Trade, Industry & Energy, Rep. of Korea, Grant/Award Number: 20174010201620.

This paper proposes a human-like sign-language learning method that uses a deep-learning technique. Inspired by the fact that humans can learn sign language from just a set of pictures in a book, in the proposed method, the input data are pre-processed into an image. In addition, the network is partially pre-trained to imitate the preliminarily obtained knowledge of humans. The learning process is implemented with a well-known network, that is, a convolutional neural network. Twelve sign actions are learned in 10 situations, and can be recognized with an accuracy of 99% in scenarios with low-cost equipment and limited data. The results show that the system is highly practical, as well as accurate and robust.

KEYWORDS

CNN, deep learning, sign language

1 | INTRODUCTION

Sign language is essential for persons who are hearing impaired, and it is rarely used by people without this disability. Thus, persons without this disability who are not accustomed to sign language often require professional help in order to communicate with persons with a hearing impairment. However, the ability to automatically recognize sign language would enable persons without this disability to easily communicate with those who are hearing impaired. As a result of this social need, there have been several recent studies on the automatic recognition of sign language using deep-learning techniques [1].

For effective sign-language recognition, previous studies have addressed the following three difficulties [2–19]. First, the required data size is large because the data for one sign action usually consists of dozens of images. Second, there are few feature points for representing sign actions in image data. Compared to an entire image size, the area that includes the hands, which has important feature information, is relatively small. Finally, the preparation of sign-language training data sets is time consuming and costly. Insufficient data can be a big hurdle to initially augment the performance of deep learning.

Previous studies have addressed these issues by using special cameras, haptic devices, or heuristic data-augmentation processes to obtain more information, such as depth image and hand-position trajectories. In other words, some high-level features have been predefined by experts and extracted by special devices. Despite the excellent performance results, these approaches still have disadvantages with respect to flexibility and cost because it is not easy to expand the training data set with the special devices; they are also relatively expensive. In addition, the learning direction may be biased by the heuristic data modification.

In this study, to solve the problems mentioned above, human-learning processes for sign language are imitated as much as possible. First, a person can identify the meaning of a sign action after watching less than 10 sequential screen shots of the action from a manual, as shown in Figure1. In other words, it is not necessary to see all of the video frames or depth frames of the sign action. Therefore, in this study, training data are pre-processed in the same way as the images in sign language manuals. The image

This is an Open Access article distributed under the term of Korea Open Government License (KOGL) Type 4: Source Indication + Commercial Use Prohibition + Change Prohibition (http://www.kogl.or.kr/info/licenseTypeEn.do).
1225-6463/$ © 2018 ETRI

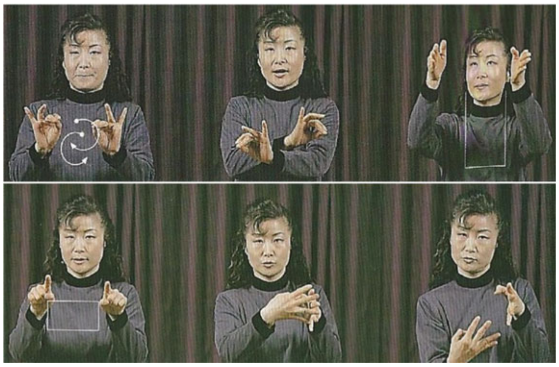


FIGURE 1 Example images in a sign language manual [20]

frames are sampled from a video clip and concatenated into an image to be used as one of the training data. Second, to identify a sign action, a person does not have to experience all of the variations, such as different people, backgrounds, and costumes. This is because people have already learned how to distinguish objects from an image. In this study, to implement this kind of prior knowledge, some general features are pre-trained. An object classification network is pre-trained using a well-known image data set published on the Internet. The weight values of this network are then transferred to initialize the sign-language learning network. As a result, the proposed sign-language learning system does not require special equipment or an algorithm to extract the hand motion. In addition, the training data size required for the same degree of capability can be reduced.

To verify the effectiveness of the proposed method, repeated tests were performed by varying the size of the pre-processed training data set. In addition, the results of the proposed method are compared to those of the method without the pre-training.

This paper is organized as follows. Sections 2 and 3 explain the related works and proposed methodology in detail, respectively. In Section 4, the experimental results are demonstrated. Finally, Section 5 presents conclusions.

2 | RELATED WORKS

Sign language involves the use of gestures. Thus, this section introduces gesture-recognition methods, which can be divided into two categories, namely hand-gesture recognition [2–19,21] and non-hand-gesture recognition [22,23]. Because the proposed method is a hand-gesture recognition method, this section introduces several competitive approaches from among the developed hand-gesture recognition methods. In addition, they are compared with the proposed method in Section 4.

Oliveira and others used a grayscale camera, convolutional neural network (CNN), and principal component analysis (PCA) to recognize Irish sign language [13]. The network consisted of four convolutional layers and a full-connected layer. As the classes, 23 Irish sign spells were used. The training data were collected by six persons in front of a fixed background by performing more than 4,000 repeated operations for each sign. The final test accuracy was 98%. Note that the 24 fingerspelling classes are not dynamic motions, but static postures.

Cooper and others adopted Microsoft Kinect as the image sensor [2]. This sensor can obtain depth information, as well as RGB information. With this information, appearance-based hand features are extracted and both the two-dimensional (2D) and three-dimensional (3D) positions of the hand can be tracked. Finally, gestures are classified using a hidden Markov model and sequential pattern boosting. Twenty Greek sign-language actions were used as classes. The training data were collected by seven persons in front of a fixed background by performing seven repeated operations for each sign action. The final test accuracy was 76%.

Wu and others used Microsoft Kinect to obtain RGB, depth, and skeleton information [3]. The RGB and depth frames were trained using CNN, and the skeleton frames were trained using the deep-belief network (DBN). Twenty sign actions were used as classes, and the training data consisted of 940 video clips. The final test accuracy was 86.4%.

3 | PROPOSED METHOD

For human-like sign-language learning, unlike the previous studies, there are two important procedures in the proposed method: data pre-processing and general feature pre-training. Both of these procedures are explained in detail below, along with the entire sign-language learning process.

3.1 | Data pre-processing

Individuals can learn sign language by reading manual in which each sign action is described using several pictures. Inspired by this approach, in the proposed method, one set of input data is a single image made from the corresponding video clip. From the video clip, K frames are sampled with interval I and serially concatenated into an image. The flow diagram of the data pre-processing is described in Figure 2.

In this paper, the number of sampled frames K is set to 9. Each image data size increases with increasing K, and there exists an upper limit of K according to the GPU memory size. For our graphic device, the maximum value of K is 9.

In addition, the sampling interval I is set to 3 and 4. According to our data from an over 60-min-long video

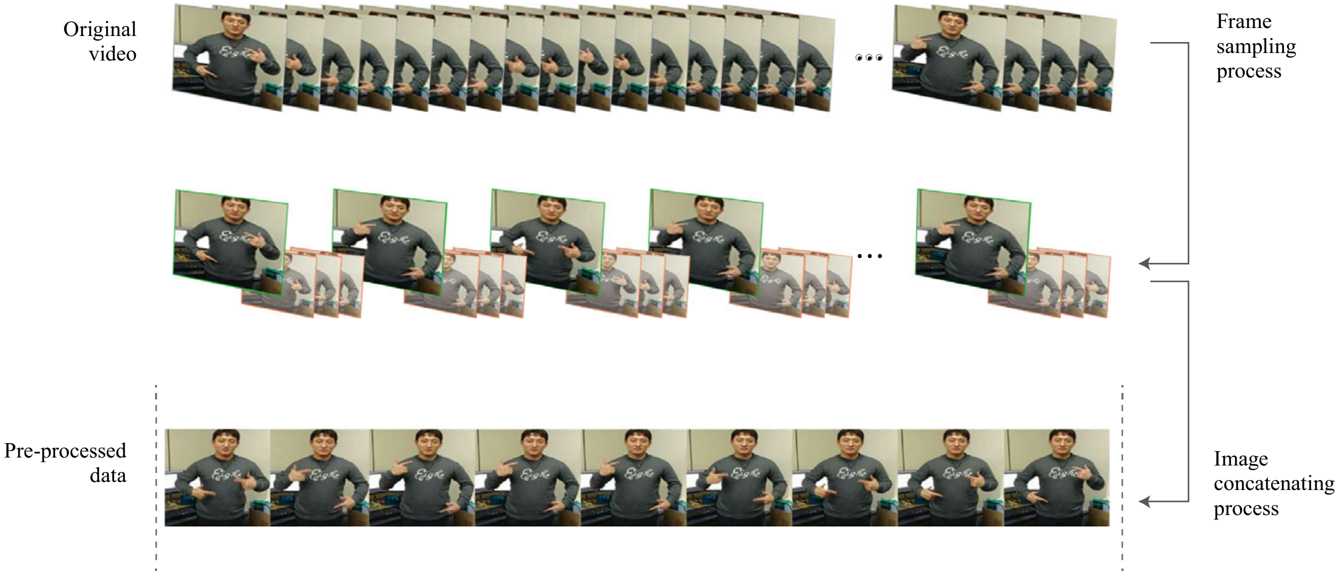


FIGURE 2 Flow diagram of the data pre-processing

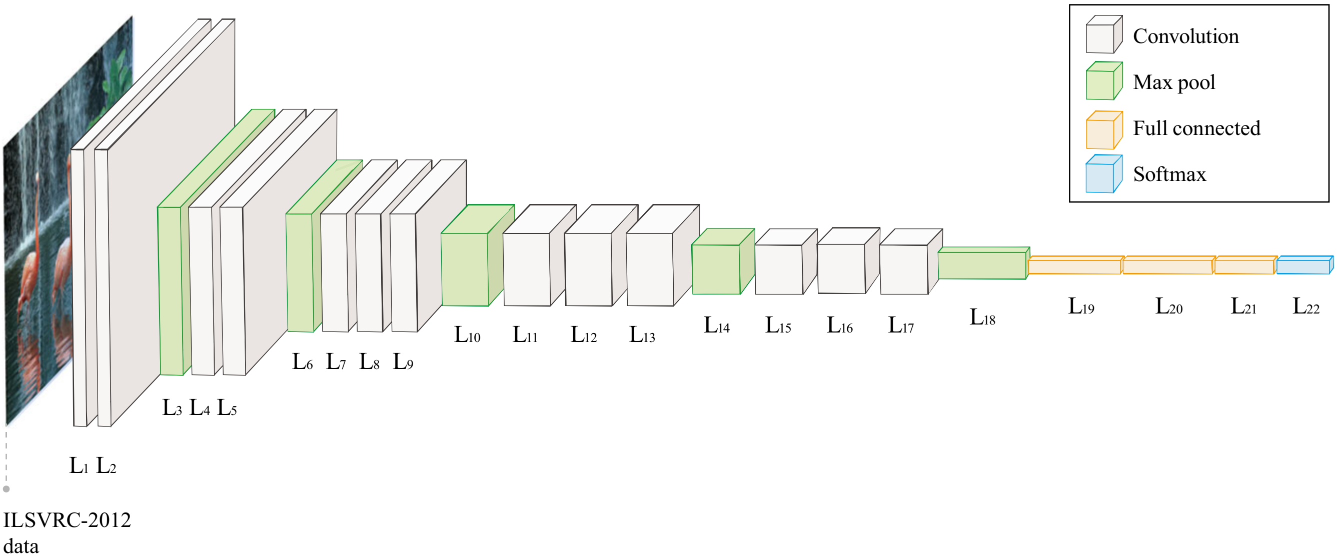


FIGURE 3 Structure of general feature pre-training network G_p

recording by six people, the average duration of one sign action was 1.12 s, with a standard deviation of 0.33 s. Based on these statistical values, it is assumed that the time required for one performance is between $(1.12 - 2 \times 0.33, 1.12 + 2 \times 0.33) = (0.46, 1.78)$. Because the camera has a shutter speed of 30 fps and K is set to 9, the corresponding I can be 2, 3, 4, or 5. Among the possible values of I, 3 and 4 are chosen empirically. Note that each video clip is converted into two images for both $I = 3$ and $I = 4$.

3.2 | General feature pre-training

Persons possess prior knowledge related to the recognition of low-level features. For example, there is general knowledge about certain information related to the contours that distinguish objects from backgrounds. To imitate this kind of knowledge, in the proposed method, the general feature pre-training network G_p is constructed and trained beforehand as an object classifier, using a well-known public data set, ILSVRC-2012 [24]. In fact, the pre-training is not directly correlated with sign language, as every image in ILSVRC-2012 is completely unrelated to sign language. Nonetheless, object classification and sign language have several properties in common. It is necessary that a certain region of interest be recognized from the background of each image. Moreover, if we recall our infancy, the behaviors of repeatedly recognizing various objects are useful in further developing complex visual discrimination ability.

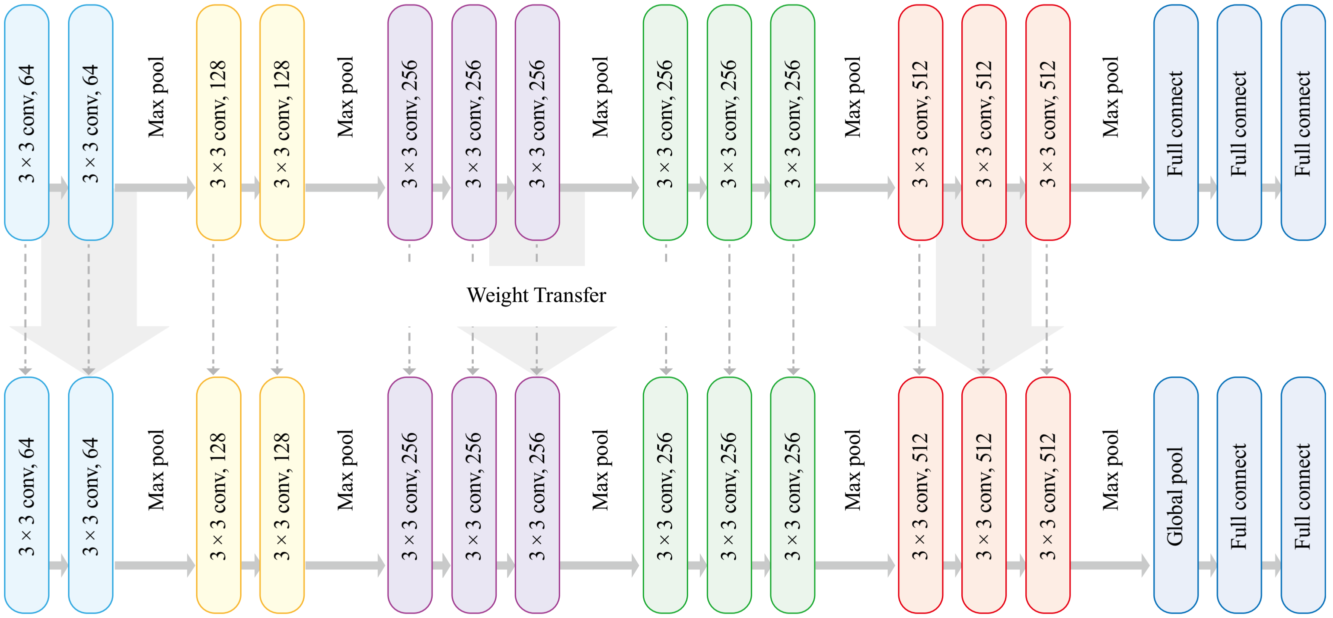


FIGURE 4 Weight transfer process

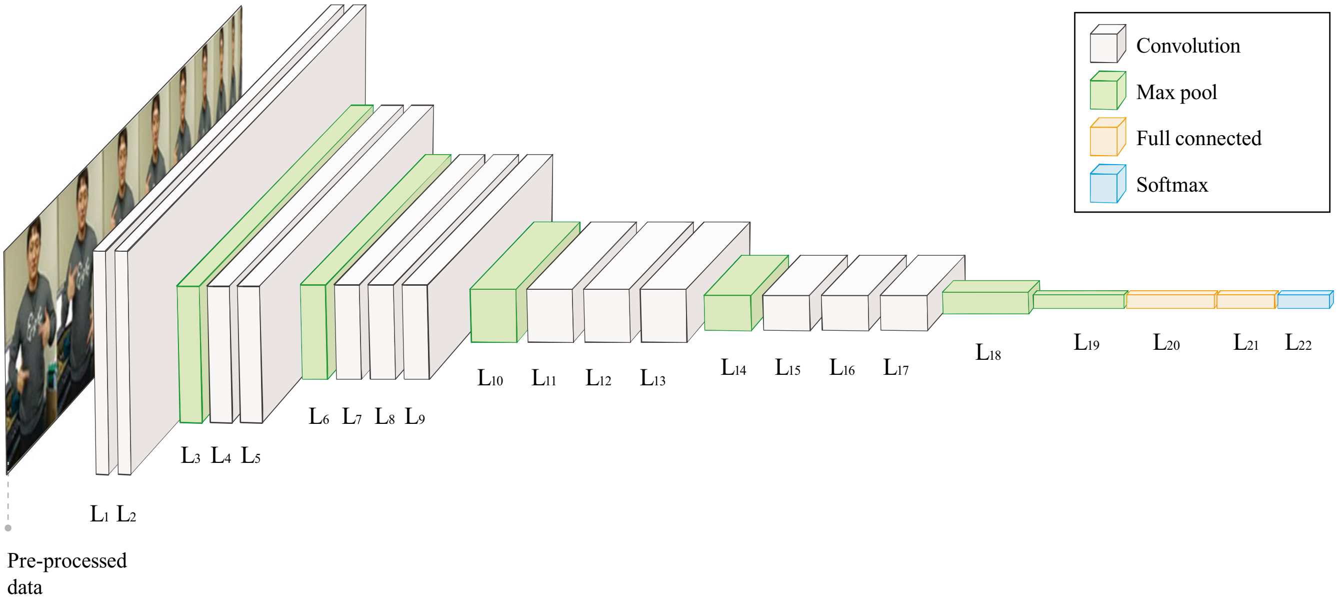


FIGURE 5 Structure of sign-language learning network G_l

As shown in Figure 3, the structure of G_p is based on a well-known network, VGG-16 [25], and consists of 22 layers, including 13 convolution layers, 5 max pool layers, 3 full connect layers, and a softmax layer. The convolution layers use 3 × 3 filters and the max pool layers use a 2 × 2 filter. The full connect layers use the activation function as follows:

$$\mathrm{ReLu}(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (1)$$

where x is the input, and the notation ReLU refers to the rectified linear unit. The final softmax layer selects one from the inputs. In G_p, each filter of the convolution layers perceives the local features, and each pool layer combines and extracts those features. Thus, the first part of the network is related to the low-level features, and the latter part is related to the high-level features. In other words, we can utilize the first part of G_p as prior knowledge.

3.3 | Weight transfer

To utilize the knowledge obtained from the pre-training, it is necessary to clarify the region of the pre-

TABLE 1 Structures of G_{p} and G_{l}

	General feature pre-training network G_{p}			Sign-language learning network G_{l}		
Layer	**Type**	**Filter size**	**Input size**	**Type**	**Filter size**	**Input size**
1	Conv	3 × 3 × 64	224 × 224 × 3	Conv	3 × 3 × 64	128 × 1,152 × 3
2	Conv	3 × 3 × 64	224 × 224 × 64	Conv	3 × 3 × 64	128 × 1,152 × 64
3	Max pool	2 × 2	224 × 224 × 64	Max pool	2 × 2	128 × 1,152 × 64
4	Conv	3 × 3 × 128	112 × 112 × 64	Conv	3 × 3 × 128	64 × 576 × 64
5	Conv	3 × 3 × 128	112 × 112 × 128	Conv	3 × 3 × 128	64 × 576 × 128
6	Max pool	2 × 2	112 × 112 × 128	Max pool	2 × 2	64 × 576 × 128
7	Conv	3 × 3 × 256	56 × 56 × 128	Conv	3 × 3 × 256	32 × 288 × 128
8	Conv	3 × 3 × 256	56 × 56 × 256	Conv	3 × 3 × 256	32 × 288 × 256
9	Conv	3 × 3 × 256	56 × 56 × 256	Conv	3 × 3 × 256	32 × 288 × 256
10	Max pool	2 × 2	56 × 56 × 256	Max pool	2 × 2	32 × 288 × 256
11	Conv	3 × 3 × 512	28 × 28 × 256	Conv	3 × 3 × 512	16 × 144 × 256
12	Conv	3 × 3 × 512	28 × 28 × 512	Conv	3 × 3 × 512	16 × 144 × 512
13	Conv	3 × 3 × 512	28 × 28 × 512	Conv	3 × 3 × 512	16 × 144 × 512
14	Max pool	2 × 2	28 × 28 × 512	Max pool	2 × 2	16 × 144 × 512
15	Conv	3 × 3 × 512	14 × 14 × 512	Conv	3 × 3 × 512	8 × 72 × 512
16	Conv	3 × 3 × 512	14 × 14 × 512	Conv	3 × 3 × 512	8 × 72 × 512
17	Conv	3 × 3 × 512	14 × 14 × 512	Conv	3 × 3 × 512	8 × 72 × 512
18	Max pool	2 × 2	14 × 14 × 512	Max pool	2 × 2	8 × 72 × 512
19	Full connect	ReLU	7 × 7 × 512	Global max pool	4 × 36	4 × 36 × 512
20	Full connect	ReLU	1 × 1 × 4,096	Full connect	ReLU	1 × 1 × 512
21	Full connect	ReLU	1 × 1 × 4,096	Full connect	ReLU	1 × 1 × 512
22	Softmax	N/A	1 × 1 × 1,000	Softmax	N/A	1 × 1 × 12

TABLE 2 Data labeling

Meaning	Label	One-hot vector
(a) Love	1	{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
(b) Thank	2	{0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
(c) Happy	3	{0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}
(d) Effort	4	{0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0}
(e) Regrettable	5	{0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0}
(f) Give	6	{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}
(g) Apologize	7	{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0}
(h) Stuffy	8	{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0}
(i) Same	9	{0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0}
(j) Learn	10	{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0}
(k) Move	11	{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0}
(l) Funny	12	{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1}

trained network to be transferred. In the problem of sign-language learning, the necessary information is about visual features stored in the filters of the convolution layers. Therefore, in the weight-transfer process, all of the weight parameters in the filters of the convolution layers are copied to the corresponding position in the sign-language learning network G_{l}, as shown in Figure 4.

3.4 | Sign-language learning

After the filters of convolution layers in G_{l} are initialized as with those in G_{p}, the other layers are initialized. As shown in Figure 5, the structure of G_{l} is also based on VGG-16, and consists of 22 layers, including 13 convolution layers, 6 max pool layers, 2 full connect layers, and a softmax layer. Note that unlike G_{p}, one full connect layer is replaced by one max pool layer. This is because the number of classes is not high. Including this difference, the detailed structures of G_{p} and G_{l} are demonstrated in Table 1.

In sign-language learning, the cost C is calculated as follows:

$$C = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log p_n + (1 - y_n)\log(1 - p_n)] \quad (2)$$

where N is the length of the one-hot encoded label vector, y_n is the n-th element of the label vector, and p_n is the n-th

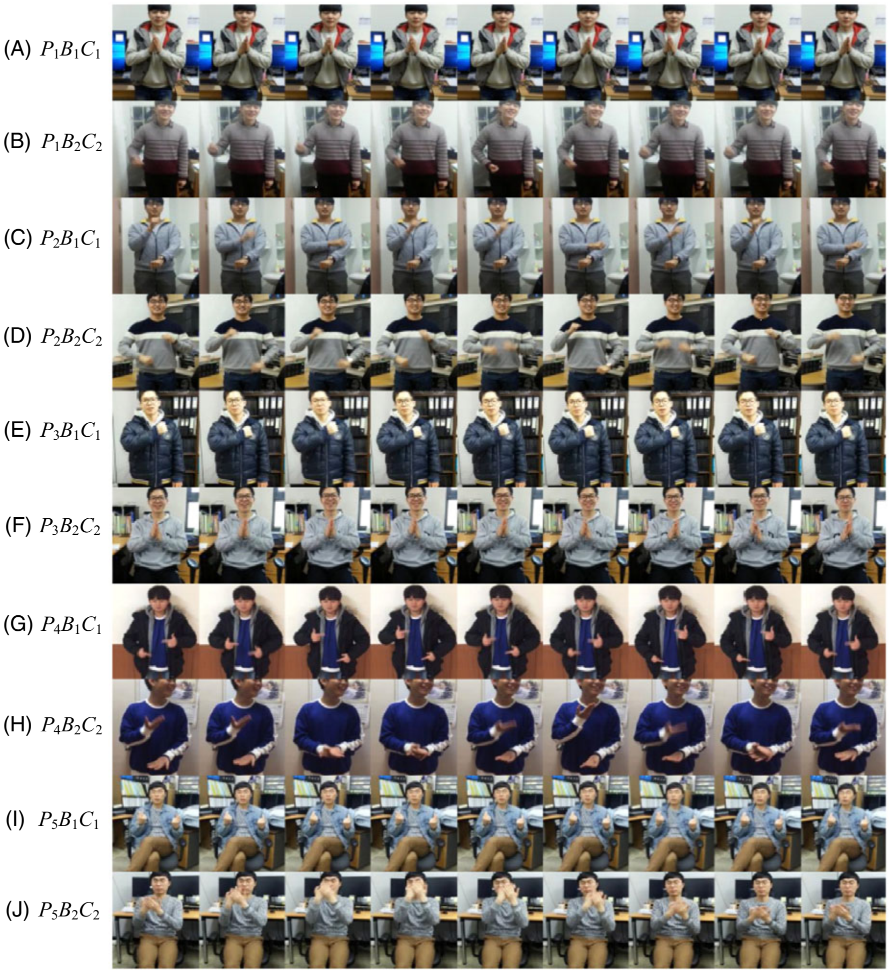


FIGURE 6 Training data examples for 10 cases

element of the probability vector p. Then, the weights of the network are updated using the back-propagation method with the cost. Likewise, the update is repeated for each input, and the process is terminated after a specified number of iterations.

4 | EXPERIMENT

In this section, detailed information about the environment settings and data set is presented. Then, the experimental results and their analysis are discussed.

4.1 | Environment settings and data set

The proposed method was implemented as code written using Python (version 3.5.2) with Tensorflow (version 1.4.1) and Keras (version 2.1.2) framework libraries. The software is run on Linux OS (version 16.04) with an Intel i7-6900K CPU, 128-GB DDR4 RAM, and an NVIDIA Titan X Pascal GPU.

As the optimizer of the neural network, the Adam algorithm [26] was adopted with a learning rate of 0.0001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The data batch size was set to 16, which is a commonly used batch size in image-processing problems. The data normalization and network initialization methods that were employed were the batch normalization method [27] and He normal initializer [28], respectively.

As the data classes, 12 Korean sign-language actions were adopted. They were labeled from 1 to 12 and represented in the one-hot vector format, as shown in Table 2.

The training data set has been collected in various scenarios, and includes combinations of different persons, backgrounds, and costumes. For each class, five different people participated in the process to generate the data. For each person, four different backgrounds and three different costumes were used. As a result, the total number of scenarios is $4 \times 5 \times 3 = 60$. In addition, for each case, the number of images is set to 50. For instance, if the number of classes is 12 and the number of scenarios is 10, then there is a total of 6,000 sets of image data.

Examples of 10 scenarios of the training data are shown in Figure 6, where P refers to a person, B represents the background, and C represents the costume.

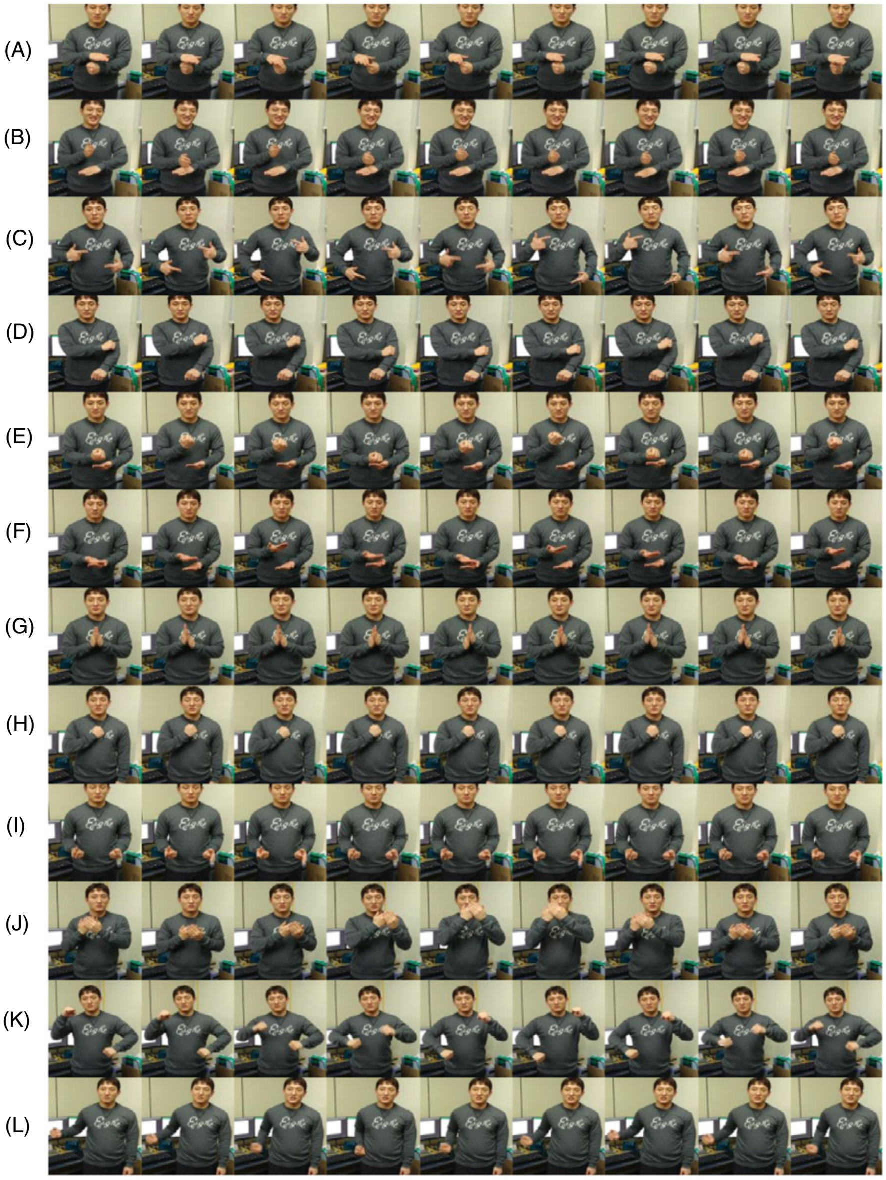


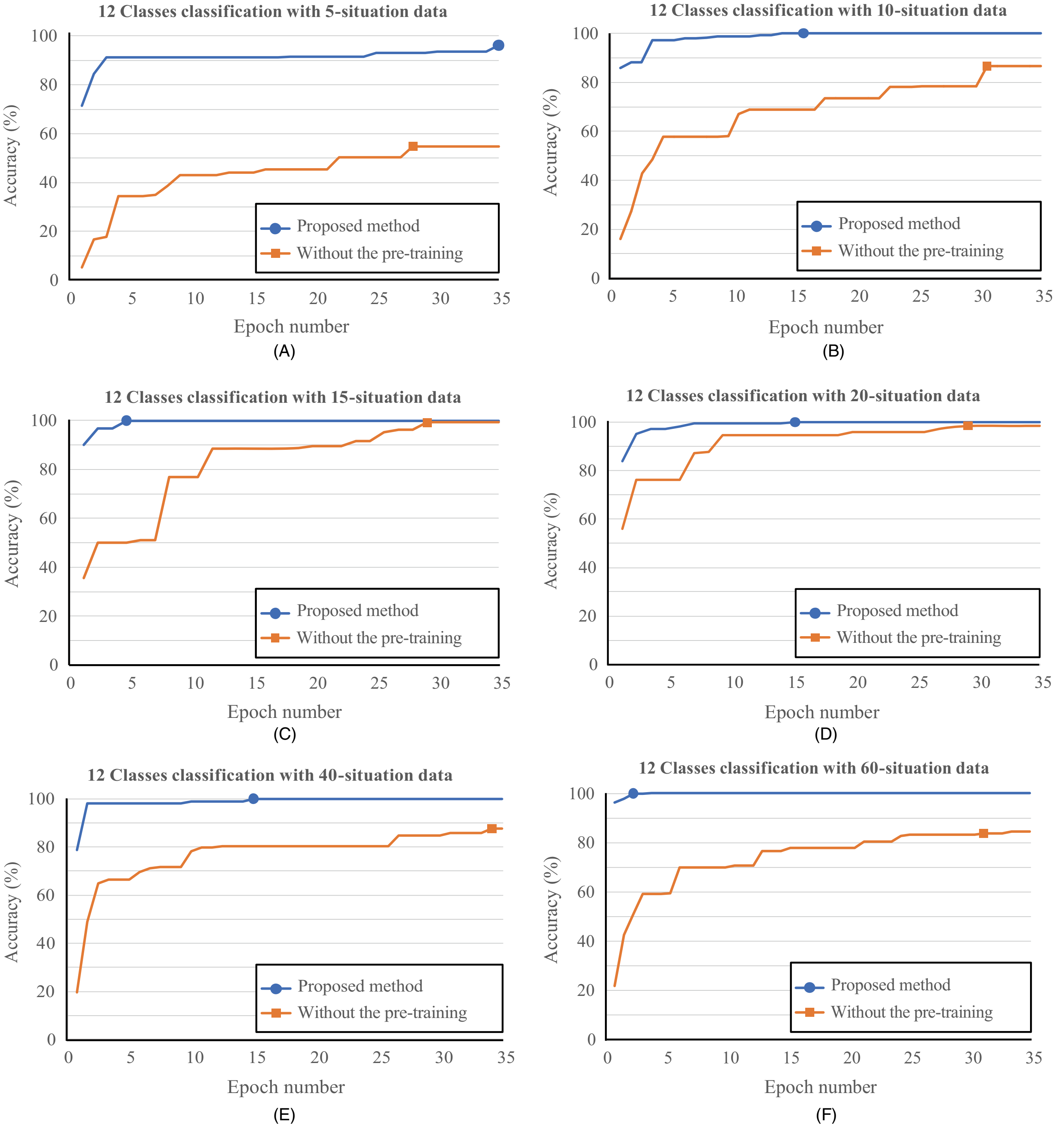
FIGURE 7 Test data examples of 12 classes

The test data set was collected in a case that is totally different from all of the situations in the training data set. The test data examples for 12 sign actions can be seen in Figure 7. Note that some classes have very similar appearances, as shown in Figure 7E and F.

4.2 | Experimental results

The experiment was performed as follows. First, the data set for the sign-language learning was pre-processed. Thereafter, G_p was trained as an image classifier using ILSVRC-2012 data. Then, the parameters of the convolution layers were transferred from G_p to G_l. Finally, G_l was trained with the pre-processed data. To demonstrate the effectiveness of the proposed pre-training process, G_l was repeatedly trained by varying the size of the data set and the test results with and without the pre-training, and the results were compared. Note that there were only 50 images in each case for the training data set after the data pre-procession.

Figure 8 shows the comparison result with the training data set, including cases with 5, 10, 15, 20, 40, and 60 scenarios. In the case with 60scenarios, the corresponding confusion matrices were also compared, as described in Table 3. Without the pre-training, the test accuracy became unstable. In other words, as shown in Figure 8E and F, the test accuracy (the orange line with square dots) could not exceed 90%, as the number of scenarios was increased to over 40. In addition, the required number of epochs for the maximum accuracy was increased. The greater the number of training data sets, the better the test accuracy in most cases. However, in the case of a tightly



(A) (B) (C) (D) (E) (F)

FIGURE 8 Comparison results: (A) 5–situation data, (B) 10-situation data, (C) 15-situation data, (D) 20-situation data, (E) 40-situation data, and (F) 60-situation data

limited sized training data, the trained network with the data could not effectively reflect the complex and high-dimensional features of the data. However, the test accuracy of the proposed method (the blue line with circle dots) exceeded 90% in all cases. Note that it took just four epochs to reach over 95%, with only data for 10 scenarios. Moreover, the accuracy quickly reached nearly 100% with data for 60 scenarios. This means that the pre-trained features could effectively help the sign-language learning network to achieve the complex and high-dimensional features of the data.

The proposed method has also been compared to the previous approaches mentioned in Section 2. As shown in Table 4, the proposed method has demonstrated

TABLE 3 Comparison of confusion matrix result with data for 60 scenarios

		Classification (data for 60 scenarios)																							
		Proposed method												**Without pre-training**											
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Ground truth	1	90												90											
	2		90												59				31						
	3			90												90									
	4				90												90								
	5					90												90							
	6						86						4						90						
	7							88				2								90					
	8								90									2			86	2			
	9									90							36					54			
	10										90												90		
	11											90											48	42	
	12												90												90

TABLE 4 Comparison of competitive methods

		Data set information				Training data vs. Test data		
Method	**Camera**	**Action**	**Classes**	**Situations**	**Images**	**Person**	**Background**	**Accuracy (%)**
Oliveira	RGB	Static	26	6	25,000	Not different	Not different	76
Cooper	RGB-D	Dynamic	20	49	N/A	Different	Not different	76
Wu	RGB-D	Dynamic	20	47	400,000	Different	Different	86.4
Proposed	**RGB**	**Dynamic**	**12**	**10**	**6,000**	**Different**	**Different**	**99**

competitive results only with an RGB camera and less data. In summary, in the proposed method, the data pre-processing could effectively extract the size of the training data, and the general feature pre-training could successfully enhance the accuracy and speed of the learning, even with less data.

5 | CONCLUSION

In this paper, a human-like sign-language learning method was proposed based on convolutional neural networks. In the proposed method, the input data were pre-processed into an image inspired from the fact that humans can learn sign language from observing several pictures in a manual. In addition, by imitating the first obtained knowledge of humans, the pre-trained network was employed to recognize the general features of an image. The experimental results showed that using the proposed method, 12 classes of the sign actions could be recognized with an accuracy of 99%. Most importantly the proposed method good potential for practical use, considering that the data set was constructed by a low-cost RGB camera and that the size of the data set was limited to 500 images per class.

ACKNOWLEDGEMENTS

This work was supported by the "Research Grant of Kwangwoon University" in 2016 and "Human Resources Program in Energy Technology" of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), and received financial support from the Ministry of Trade, Industry & Energy, Rep. of Korea (No. 20174010201620).

ORCID

Ki-Baek Lee http://orcid.org/0000-0002-3416-9176

REFERENCES

1. H. Liu et al., *Gesture recognition for human-robot collaboration: A review*, Int. J. of Ind. Ergon. 2017.

2. H. Cooper et al., Sign language recognition using sub-units, *J. Mach. Learn. Res.* **13** (2017), no. 1, 2205–2231.
3. D. Wu et al., *Deep dynamic neural networks for multimodal gesture segmentation and recognition*, IEEE Trans. Pattern Anal. Mach. Intell. **38** (2016), no. 8, 1583–1597.
4. J. Huang et al., *Sign language recognition using 3d convolutional neural networks*, IEEE Int. Conf. Multimed. Expo, Turin, Italy, June29–July 3, 2015, pp. 1–6.
5. D. Wu et al., *Deep dynamic neural networks for gesture segmentation and recognition*, Workshop Eur. Conf. Comput. Vis. **38** (2014), no. 8, 1583–1597.
6. C. Xiujuan et al., *Sign language recognition and translation with Kinect*, IEEE Int. Conf. on AFGR **655** (2013).
7. A. Agarwal et al., Sign language recognition using Microsoft Kinect, *IEEE Int. Conf. Contemp. Comput.*, Noida, India, Aug. 8–10, 2013, pp. 181–185.
8. I. Lim et al., Sign-language recognition through gesture & movement analysis (SIGMA), *DLSU Res. Congress 2*, Manila, Philippine, Mar. 2–4, 2015, p. HCT -I-011.
9. L. Pigou et al., Sign language recognition using convolutional neural networks, *Workshop Eur. Conf. Comput. Vis.*, Zurich, Swiss, Sept. 6–12, 2014, pp. 572–578.
10. O. Koller et al., Deep Sign: Hybrid CNN-HMM for continuous sign language recognition, *Proc. Br. Mach. Vis. Conf.*, York, UK, Sept. 19–22, 2016.
11. N. Neverova et al., Multi-scale deep learning for gesture detection and localization, *Workshop Eur. Conf. Comput. Vis.*, Zurich, Swiss, Sept. 6–12, 2014, pp. 1–17.
12. O. Koller et al., *Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers*, Comput. Vis. Image Underst. **141** (2015), 108–125.
13. M. Oliveira et al., A comparison between end-to-end approaches and feature extraction based approaches for sign language recognition, *Int. Conf. on Image and Vis. Comput. New Zealand*, Christchurch, New Zealand, Dec. 4–6, 2017, pp. 1–5.
14. Y. L. Gweth et al., Enhanced continuous sign language recognition using PCA and neural network features, *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recogn. Workshops*, Providence, RI, USA, June 16–21, 2012, pp. 55–60.
15. J. Forster et al., Improving continuous: Speech recognition techniques and system design, *Workshop Speech Lang. Process. Assist. Technol.*, Grenoble, France, Aug. 21–22, 2013, pp. 41–46.
16. S. Jain et al., *Indian sign language gesture recognition*, 2015.
17. A. K. Sahoo et al., *Sign language recognition: state of the art*, ARPN J. Eng. Applicat. Sci. **9** (2014), 116–134.
18. J. Singha et al., *Recognition of Indian sign language in live video*, arXiv: 1306–1301, 2013.
19. B. Garcia et al., *Real-time American sign language recognition with convolutional neural networks*, Convolutional Neural Netw. for Vis. Recogn. (2016), 225–232.
20. O. Kang, *Sign language (The most valuable language in the world)* , Light and Fragrance, 2001.
21. J.-Y. Lee et al., *A real-time hand gesture recognition technique and its application to music display system*, J. Autom. Contr. Eng. **4** (2016), no. 2, 177–180.
22. O. Koller et al., Deep learning of mouth shapes for sign language, *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Santiago. Chile, Dec. 7–13, 2015, pp. 477–483.
23. D. Weinland et al., *Free viewpoint action recognition using motion history volumes*, Comput. Vis. Image Underst. **104**, (2006) 249–257.
24. O. Russakovsky et al., *Imagenet large scale visual recognition challenge*, Int. J. of Comput. Vis. **115** (2015), 211–252.
25. K. Simonyan et al., *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv: 1409.1556, 2014.
26. D. Kingma et al., *A method for stochastic optimization*, arXiv preprint arXiv: 1412.6980, 2014.
27. S. Ioffe et al., *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, arXiv preprint arXiv: 1502.03167, 2015.
28. K. He et al., Deep residual learning for image recognition, *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, Las Vegas, NV, USA, June 27–30, 2016, pp. 770–778.

AUTHOR BIOGRAPHIES

Yangho Ji received his BS degree in electrical engineering from Kwangwoon University, Seoul, Rep. of Korea, in 2016. Since 2016, he has been pursuing his MS degree in the Department of Electrical Engineering, Kwangwoon University. His research interests include computer vision and machine learning.

Sunmok Kim received his BS degree in electrical engineering from Kwangwoon University, Seoul, Rep. of Korea, in 2016. Since 2016, he has been pursuing his MS degree in the Department of Electrical Engineering, Kwangwoon University. His research interests include reinforcement learning and machine learning.

Young-Joo Kim received his BS degree in mechanical engineering, and his MS and PhD degrees in industrial engineering in 1999, 2002, and 2008, respectively, from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea. He has conducted research on logistics-related technology as a senior researcher at the Korea Railroad Research Institute (KKRI) from 2007 to 2017. Currently, he is a principal researcher in the Logistics System Research Team at KRRI. His research interest is intelligent logistics automation.

Ki-Baek Lee received his BS and PhD degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2005 and 2014, respectively. Since 2014, he has been an assistant professor with the Department of Electrical Engineering, College of Electronics and Information Engineering, Kwangwoon University, Seoul, Rep. of Korea. He has researched computational intelligence and artificial intelligence, particularly in the area of swarm intelligence, multiobjective evolutionary algorithms, and machine learning. His research interests include real-world applications such as AI sign-language recognition for IoT devices.