IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 9, 2020, accepted March 5, 2020, date of publication March 11, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980128

Skeleton-Based Dynamic Hand Gesture Recognition Using a Part-Based GRU-RNN for Gesture-Based Interface

SEUNGHYEOK SHIN AND WHOI-YUL KIM
Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Whoi-Yul Kim (wykim@hanyang.ac.kr)

This work was supported in part by the Korea Evaluation Institute of Industrial Technology (KEIT) and Granted Financial Resources by the Korean Government (Development of HD Resolution and mm-Level Precision 3D Gesture Recognition Camera for Attention and Convenience Maximization) under Grant 10080639, and in part by the Human Resources Program of Korea Institute of Energy Technology Evaluation and Planning (KETEP) and Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea (Human resource development for R&D of Nuclear decommissioning) under Grant 20184030201970.

ABSTRACT Recent improvements in imaging sensors and computing units have led to the development of a range of image-based human–machine interfaces (HMIs). An important approach in this direction is the use of dynamic hand gestures for a gesture-based interface, and some methods have been developed to provide real-time hand skeleton generation from depth images for dynamic hand gesture recognition. Towards this end, we propose a skeleton-based dynamic hand gesture recognition method that divides geometric features into multiple parts and uses a gated recurrent unit-recurrent neural network (GRU-RNN) for each feature part. Because each divided feature part has fewer dimensions than an entire feature, the number of hidden units required for optimization is reduced. As a result, we achieved similar recognition performance as the latest methods with fewer parameters.

INDEX TERMS Artificial neural networks, gesture recognition, multi-layer neural network, recurrent neural networks.

I. INTRODUCTION

Gestures are the basic elements used by humans to express meaningful movements [1], and many studies have been conducted on the development of gesture-based human–machine interfaces (HMIs) [2]. Hand gestures are natural and frequently used in face-to-face interactions; therefore, they can be used to make intuitive HMIs [3]. Although some researchers have used ''data gloves'' to acquire hand movement information [4], this method cannot be widely implemented because it requires expensive hardware. Thus, recent studies have proposed hand gesture recognition using image-based methods incorporating relatively cheap imaging sensors.

Hand gestures can be either static or dynamic [5]. Static hand gestures are represented by the hand's shape, and, therefore, complex hand poses may be required to represent many types of static hand gestures. In contrast, dynamic hand gestures involve both hand shape and movement, and, therefore, the limitation imposed by hand poses is less restrictive than with static hand gestures. Owing to developments and improvements in three-dimensional (3D) sensors, hand gesture recognition can be conducted using not only hand shapes and movements but also a hand skeleton. A hand skeleton for gesture representation comprises a connection between all joints connecting neighboring phalanges. If a hand pose changes for a dynamic hand gesture, we can predict which gesture is being performed from the hand skeleton. To exploit this property, many researchers have extracted features from hand skeletons [6]–[8] through detailed examinations. Various methods for hand feature classification have also been proposed [6], [7], [9], [10], and recent improvements in parallel computing have enabled the adoption of neural-network-based feature classification methods [7], [8], [11]–[14]. Most neural networks demonstrated better performance than non-neural-network classifiers; however, they required considerable number of parameters and high-performance hardware. Thus, constructing gesture-based HMI systems based on neural networks is hardly feasible.

The associate editor coordinating the review of this manuscript and approving it for publication was Huanqing Wang.

In this study, we propose a neural-network-based recognition method for dynamic hand gestures that is suitable for constructing HMI systems. The main contributions of our proposed method are as follows:

1) We divided features into multiple parts and provided each part as the input of a gated recurrent unit-recurrent neural network (GRU-RNN). By dividing the features, the number of dimensions of the input features as well as the number of parameters of the GRU-RNNs are reduced. Thus, relatively faster training and data recognition than existing methods are possible.
2) The output of each GRU-RNN is concatenated with other feature parts; therefore, the relation between the hand parts is conserved with fewer GRU-RNN parameters. Because of this conservation, we achieved similar recognition performance as other existing methods while using fewer parameters, and our method can be implemented even with low-performance hardware.

The proposed method recognizes dynamic hand gestures by removing noise, spatially normalizing skeleton coordinates, extracting geometric features from a hand skeleton, dividing the features into multiple parts, and classifying these features using a part-based GRU-RNN (PB-GRU-RNN). In the PB-GRU-RNN, each feature part has its own GRU-RNN, and the output is fed to the next part's GRU-RNN. Therefore, our neural network requires less memory than other neural-network-based methods that offer similar recognition performance.

The rest of this paper is organized as follows: Section II introduces existing hand gesture recognition methods and their limitations. Section III introduces and describes the proposed method. Section IV presents our experimental settings and results. Section V presents a comparison of our method with other methods and analyzes the experimental results. Finally, Section VI describes our contributions and outlines future work.

II. RELATED WORK

Because image-based hand gesture recognition can be used in areas other than HMI, it has attracted much research attention in the last 30 years [15]. Before the development of image-based 3D sensors, many hand gesture recognition methods used color information to segment a hand from the background of an image [16]–[20]. However, 3D sensors, such as Kinect, RealSense, and Leap Motion, have made hand segmentation easier by using a combination of color and depth. Therefore, methods for recognizing static and dynamic hand gestures by using 3D sensors have become popular.

Static hand gestures are represented by hand shapes, and they can be recognized by describing hand shapes and then measuring the similarity between shapes; however, the number of available gesture types is limited because complex hand shapes may be required to represent more static hand gesture types. Dynamic hand gestures suffer less from this limitation than static gestures. However, temporal behaviors, which can be represented as hand movements and transitions between static hand poses, must be dealt with.

Wu and Lin [6] recognized dynamic hand gestures by measuring transitions between static hand gestures by the region of feature and support vector machines (SVMs) in a predetermined duration. Although their method was simple to implement, only a few dynamic gestures could be recognized because the number of transitions is limited. Instead of using hand transitions, a method for describing hand movements as features and classifying them with hidden Markov models (HMMs) has been proposed. Beh *et al.* [9] used a left-to-right HMM to recognize dynamic hand gestures as represented by two-dimensional trajectories. For the HMM input, trajectories were segmented based on abrupt angle changes, and points on the trajectories were resampled to ensure that the data from each class had the same number of segments. The resampled data were classified by left-to-right HMMs that were initialized by a mixture of von Mises distribution-based HMMs (MvM-HMMs). Each segment was modeled by the MvM to describe its state. HMM-based methods could recognize dynamic gestures with decent performance; however, this was achieved using strict conditions such as the number of states, initialization methods, and feature types.

To overcome the condition limitations of HMM-based methods, Cheng *et al.* [10] adopted image-to-class dynamic time warping (I2C-DTW) to recognize dynamic hand gestures by comparing hand movement trajectories. Features were extracted from 3D hand trajectories and were classified through I2C-DTW, and they showed a better performance than HMMs. De Smedt *et al.* [7] extracted features called the shape of connected joints (SoCJ), histogram of hand directions (HoHD), and histogram of wrist rotations (HoWR) from a hand skeleton. The SoCJ was encoded by Fisher vectors, and an SVM was used to classify the combination of encoded SoCJ, HoHD, and HoWR. Their method used a hand skeleton and its joints instead of a single trajectory to represent dynamic hand gestures. Therefore, recognizing more complex hand gestures, such as pinching or grabbing, was possible. Other than typical machine-learning-based methods, some scholars have proposed the use of neural networks owing to improvements in parallel computing hardware. Molchanov *et al.* [11] used a convolutional neural network (CNN) with grayscale and depth image sequences to recognize dynamic hand gestures. The image sequences were temporally normalized to a fixed size for CNN inputs. In their study, two CNNs were used: one for the original image sequence and another for a spatially resized image sequence. De Smedt *et al.* [21] extracted neighboring keyframes, which are regularly picked from a depth image sequence, and used a CNN to classify these keyframes to recognize dynamic hand gestures. Devineau *et al.* [12] used a CNN called multichannel deep CNN (MC-DCNN), with a combination of fixed-length 1D sequences as input to recognize skeleton-based dynamic hand gestures. The multidimensional sequence of each frame was split into multiple

1D sequences, and each 1D sequence was processed by the MC-DCNN. For the fixed-length input of their CNN, the input sequence was temporally normalized. Each channel was propagated through three branches: two for feature extraction at different resolutions and one for pooling to prevent over-fitting. Chen *et al.* [8] used long short-term memory RNNs (LSTM-RNNs) and motion features to recognize skeleton-based dynamic hand gestures. They proposed two types of features: global and finger motion features. Each type of feature was processed by LSTM-RNN layers followed by fully connected layers. Zhu *et al.* [13] proposed a combination of 3D CNN and LSTM to segment continuous dynamic gestures and recognize the segmented dynamic gestures. They used two types of neural networks: segmentation and recognition network. A segmentation network determines the start and end frames by using RGB modality, depth modality, and ConvNet networks. In the recognition network, rank-pooled depth sequences, RGB sequences, and saliency sequences are fed to neural networks to recognize the gesture between the start and the end frames [22]. Maghoumi and LaViola, Jr. [14] used a deep GRU-RNN with an attention module to recognize dynamic hand gestures. They augmented the data with random scaling, random translation, and synthetic sequence generation with stochastic resampling and used only hand joints. Avola *et al.* exploited the angles between adjacent fingers and adjacent phalanges to extract better features for their LSTM-RNN network. The data were then preprocessed using the Savitzky–Golay filter to remove noise, and the timesteps with the peaks of the feature values were selected to create an input with fixed-length for their LSTM-RNN.

III. DYNAMIC HAND GESTURE RECOGNITION WITH A PART-BASED GRU-RNN

To recognize dynamic hand gestures from the 3D skeletal joint sequences of a hand, our proposed method comprises four stages: noise removal, data normalization, feature extraction, and gesture recognition. Noise removal is performed using the Savitzky–Golay filter. Data normalization spatially normalizes skeletal joint sequences, after which features are extracted from parts of the normalized sequences. The extracted features are then classified by the PB-GRU-RNN for dynamic hand gesture recognition.

In this study, we separated the hand into two parts: the fingers and the palm. We defined the finger as the combination of adjacent phalanges, and defined the palm as the combination of carpals and metacarpals except for the thumb metacarpal. The thumb is a combination of the metacarpal and phalanges and is an exception because it does not have an intermediate phalanx. Thus, each finger has three bones. Figure 1 shows the anatomy of the hand skeleton along with our definition of the palm, fingers, and hand joints. The palm contains seven joints, and each finger comprises three joints. Hereafter, the explanation for the palm and fingers is based on Figure 1.

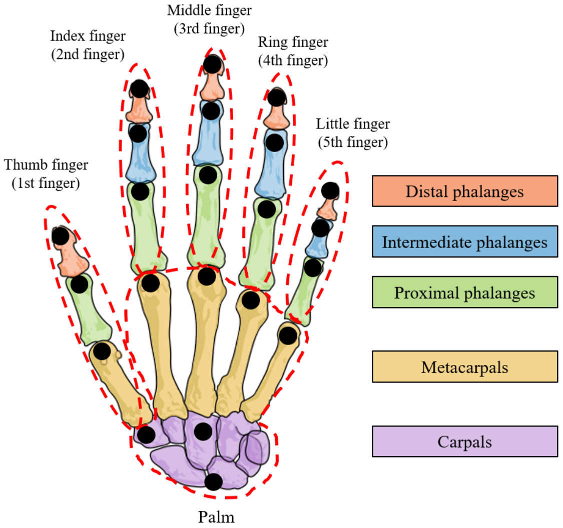


FIGURE 1. Anatomy of hand skeleton and definition of the palm, fingers (red dashed lines), and hand joints (black dots) in this study.

A. NOISE REMOVAL

Depending on the sensor and environment, the acquired joints may oscillate, and these oscillating noises may hinder the performance of hand gesture recognition. Thus, noise removal is required for better recognition performance. The Savitzky–Golay filter [23] is a polynomial-based filter that is used to smooth digital signals (*e.g.*, the coordinates of hand skeleton joints) without distorting the signal tendency, such as the positions of the extrema of the signal.

B. DATA NORMALIZATION

Hand gesture data may have the same gesture label, but the gestures' scale and speed may vary because of factors such as user behavior and physical characteristics, sensor resolution, and the distance between a sensor and the hand. In this study, we normalized our data spatially only because the time-dependent factors could be handled by an RNN.

Because the starting position of a gesture should not interfere with gesture recognition, all joints in all frames were translated with respect to the coordinates of a reference point. In this study, the wrist joint in the first frame was selected as the reference point. After translation, the initial size of hand in the sequence can be estimated because the gestures can be distinguished by the change in hand size. The initial size of the hand can be estimated by the distance between the palm center and the farthest joint in the first frame, D. All hand joint coordinates are divided by D for data normalization.

C. FEATURE EXTRACTION

The proper recognition of dynamic hand gestures requires features that describe the hand and finger movements as well as the pose. We divide these features into three parts: palm, finger, and pose.

For hand gestures, the palm is the base and other geometric features are related to the palm joints. For the palm gestures, the following features are extracted:

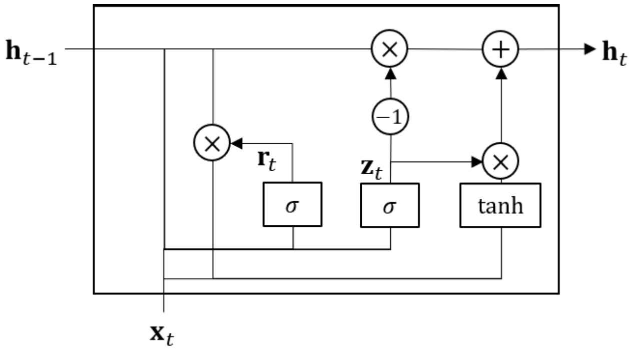


FIGURE 2. Structure of a GRU for an RNN.

- Coordinates of the palm joints.

Various hand gestures are expressed by the movement and folding of fingers, thus we can extract distinguishable features for gesture recognition from them. For the finger part, the following features are extracted, motivated by [24]:

- Relative coordinates of the finger joints and fingertip from the wrist joint.
- Internal angles between the adjacent bones.
- Vectors between adjacent finger joints for each finger.
- Internal angles between fingers as determined by the wrist–fingertip vectors.

The hand pose is determined by a combination of the palm and fingers. For the pose part, the following features are extracted:

- Palm tilting is represented by the normal of the plane generated by the wrist joint, thumb metacarpal end, and little metacarpal end.
- Movement of the palm center and fingertips.
- Wrist–palm center vector and palm center–fingertip vectors.
- Internal angles between the normal of the palm tilting plane and the palm center–fingertip vectors.

D. GESTURE RECOGNITION

An RNN is a feed-forward neural network that incorporates the temporal behavior of the input data sequence [25]. RNN optimization may be hindered by the expansion or reduction of gradients during error backpropagation when the sequence is long. We used GRUs [26], as shown in Figure 2, to overcome the gradient problem.

A GRU has two "gates," which are pairs of a vector and matrix: update gate z and reset gate r. In Figure 2, the output vectors of these gates at timestep t are denoted as z_t and r_t, respectively, and the output of the hidden layer at timestep t is denoted as h_t. These output vectors are calculated as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$
$$h_t = (1-z_t) \circ h_{t-1} + z_t \circ \tanh(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h), \tag{3}$$

where t indicates the index of timestep; $\circ$ indicates the Hadamard product operation; tanh is the hyperbolic tangent function; σ is the sigmoid function; and W_α, U_α and b_α are the weight matrix and bias of gate α shared by the hidden units in the same layer. Thus, there are $3L$ pairs of weight matrices and biases for our GRU-RNN with L layers. h_t is the output vector from the hidden unit that is delivered to the hidden unit at timestep $t+1$ in the same layer or to the hidden unit at timestep t in the next layer. As the gates' names and (3) imply, an increase in r_t decreases the weight of h_{t-1} to update h_t whereas an increase in z_t decreases the weight of h_t to reset h_t to h_{t-1}.

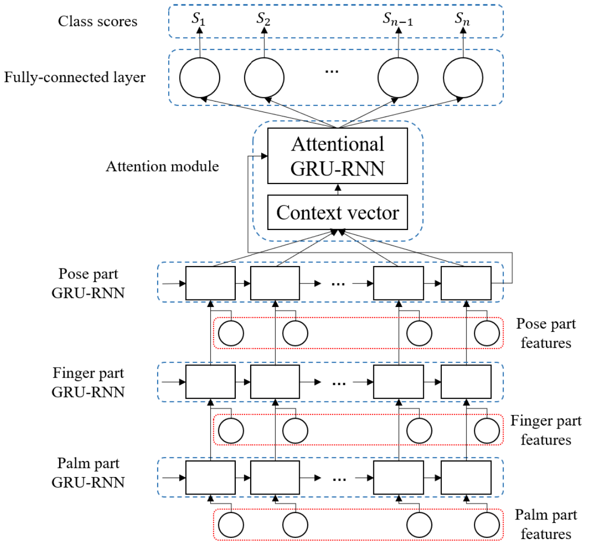


FIGURE 3. Structure of PB-GRU-RNN for dynamic hand gesture recognition.

Our GRU-RNN network, the PB-GRU-RNN, has three parts in the following order: the palm, finger, and pose parts. Each part has its own features, as described in Section III-C, and its own GRU-RNN. The output of the GRU-RNN of each part is concatenated with the features from the next part, and the concatenated data is fed to the GRU-RNN of the next part; thus, the relationships between the parts are maintained. The length (*i.e.*, number of frames) of gestures is arbitrary and, therefore, the significance of each time step may not be the same. Thus, we calculated the weight for each timestep by using the attention module proposed by Maghoumi and LaViola, Jr. [14]. In this attention module, the context vector is generated by a trainable fully connected (FC) layer. The context vector is fed to the attentional GRU-RNN, and the hidden unit of the last timestep of the GRU-RNN of the last part is given as the initial state of the attentional GRU-RNN. The output of the attentional GRU-RNN is fed to a FC layer with an activation function designed to calculate the class scores for classification. The class with the highest score is assigned to the datum. Figure 3 shows the structure of the PB-GRU-RNN for the n-class classification of a sequence of length T.

IV. EXPERIMENTAL RESULTS

A. DATASET

We used the SHREC'17 dataset [21], comprising 2,800 hand gestures created by 28 participants, for our experiments.

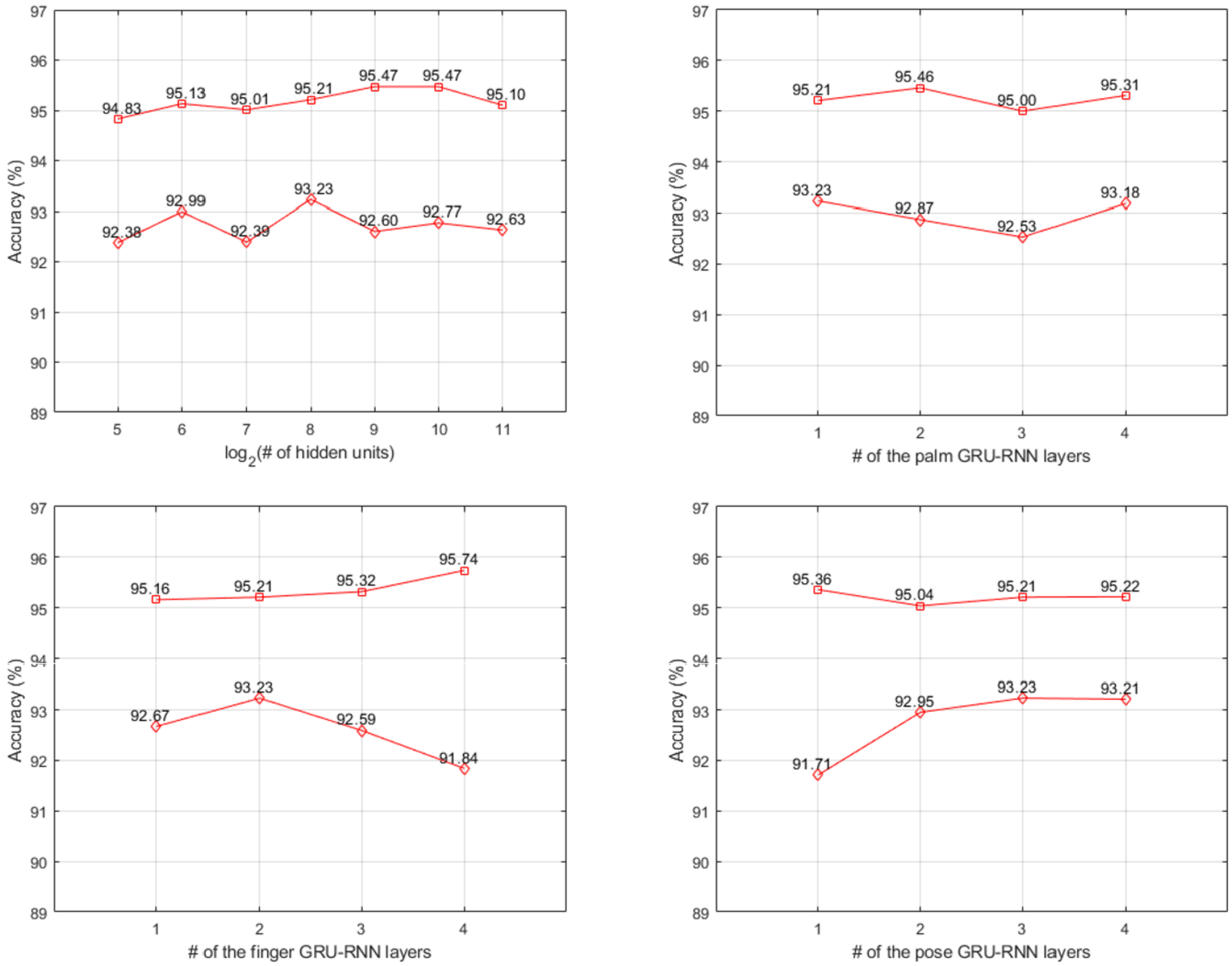


FIGURE 4. Performance (%) comparison w.r.t. hyperparameters of PB-GRU-RNN.

Each hand gesture comprises a sequence of 7–170 images captured by an Intel RealSense depth camera, and each frame of a hand gesture comprises data from 22 joints. There are 14 gesture types in the SHREC'17 dataset: *Grab*, *Tap*, *Expand*, *Pinch*, *Rotation Clockwise*, *Rotation Counterclockwise*, *Swipe Right*, *Swipe Left*, *Swipe Up*, *Swipe Down*, *Swipe X*, *Swipe +*, *Swipe V*, and *Shake*. These gesture types are called coarse gestures, each of which can be further categorized as one of two fine gestures based on finger-folding patterns. Thus, the SHREC'17 dataset comprises 14 and 28 coarse and fine gesture types, respectively. Of the SHREC'17 data, 70% (1,960 data) was used as the training dataset and the other 30% (840 data) as the test dataset.

B. EXPERIMENTAL SETTINGS

For data normalization, the joint movements in the first frame were set to 0 because they could not be calculated.

For the hidden and FC layers, the dropout ratio was set to 50%. The epoch for the training data was set to 200. Each batch consisted of 14 gesture data. We used the Adam optimization [27] method to optimize our network, where the learning rate was set to 10^{-3}, and the exponential decay rates for the first- and second-moment estimates were set to 0.9 and 0.999 respectively. Furthermore, the value for the zero-denominator prevention was set to 10^{-8}. The class scores were calculated using the FC layer with the softmax activation function, and the inputs of the FC layer were batch-normalized. All elements of h_0, the input hidden vectors for the hidden units at the first timestep, were set to 0 for all hidden layers.

As mentioned in Section III-C, we divided the features into three parts and, therefore, the hidden layers were also divided into three parts for each feature part. In our experiment, the number of hidden layers for the palm, finger, and pose parts were set to 1, 2, and 3, respectively. In addition, the number of hidden units per layer for these parts were set to 32, 64, and 64, respectively; and the number of hidden units for the FC layer was set to 256 as the basic setting. The total number of parameters was approximately 220,000.

For the attention module, we used the GRU-RNN having one layer with 256 hidden units. Moreover, for the FC layer attached to the attention module, 256 hidden units were used.

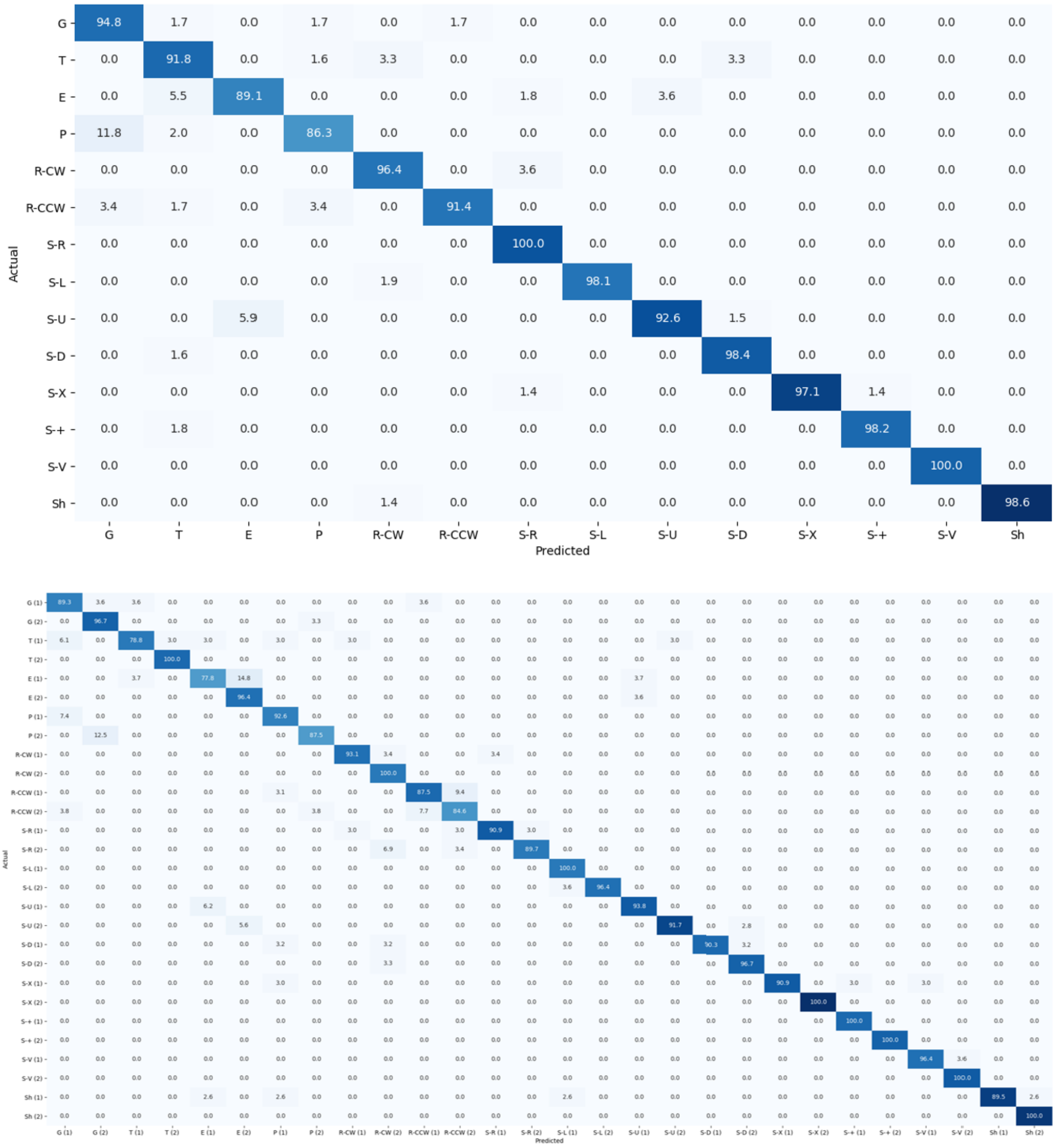


FIGURE 5. Confusion matrix for coarse gestures (top) and fine gestures (bottom) obtained by PB-GRU-RNN.

Table 1 shows the comparison between the accuracies of the existing approaches and the proposed PB-GRU-RNN.

C. ADJUSTMENT OF HYPERPARAMETERS

We also examined the change in performance when the hyperparameters of the PB-GRU-RNN were adjusted. First, we only adjusted the number of hidden units in the FC layer by multiplying or dividing by a power of 2 while other hyperparameters were fixed. We also adjusted the number of layers of the GRU-RNN of each part while other hyperparameters were fixed. The adjusted PB-GRU-RNNs were tested on the SHREC'17 dataset, and Figure 4 shows the result.

TABLE 1. Test accuracy (%) on the SHREC'17 dataset.

	Coarse	Fine
Oreifej *et al.* [28]	78.53	74.03
Ohn-Bar & Trivedi [29]	83.85	76.53
Devanne *et al.* [30]	79.61	62.00
De Smedt *et al.* [21]	82.90	71.90
De Smedt *et al.* [7]	88.24	81.90
Chen *et al.* [8]	84.68	80.32
Devineau *et al.* [12]	91.28	84.35
Maghoumi & LaViola Jr. [14]	94.50	91.40
Avola *et al.* [24]	97.62	91.43
PB-GRU-RNN	95.21	93.23

TABLE 2. Test accuracy (%) on the SHREC'17 dataset without noise removal, batch normalization, spatial normalization, or attention module.

	Coarse	Fine
Basic setting	95.21	93.23
W/o noise removal	94.99	91.70
W/o batch normalization	95.15	91.38
W/o spatial normalization	95.06	92.22
W/o attention module	94.38	86.82

D. OTHER SETTINGS

We applied noise removal, batch normalization, spatial normalization, and an attention module in our experiments and examined how much the recognition performance dropped when these methods were not used. Table 2 shows the results for these settings.

V. DISCUSSION

As shown in Table 1, our method outperformed existing non-neural-network-based methods as it did with existing neural-network-based ones.

Regarding the number of hidden units in the FC layer, the recognition performance for fine gestures peaked when the number of hidden units was 256. For coarse gestures, the recognition performance peaked when the number of hidden units were 512 and 1,024. We assumed that the recognition performance was saturated with 256 hidden units and, thus, the use of more units would not improve performance.

As shown in Figure 4, some of the adjusted hyperparameters showed slightly better performance for coarse gestures than for the basic settings; however, none of them showed the same performance for fine gestures. This could be because when some of part features were relatively nonuseful (*e.g.*, the use of the finger features for coarse gestures), the number of layers of that part had smaller impact on the recognition performance.

Noise removal, batch normalization, spatial normalization, and the attention module improved recognition performance, as shown in Figure 2. Among these four methods, the use of the attention module drastically improved the performance for fine gestures, whereas the use of the other methods only slightly improved the performance. Compared to the LSTM-RNN method proposed by Chen *et al.* [8], our method showed improved recognition accuracy for both gesture types. Their method used only motion features; we assumed that representing gestures with only motion was not enough.

The MC-DCNN method proposed by Devineau *et al.* showed decent performance by using only the coordinates of the hand joints; however, it seemed that the use of only 1D coordinates was unsuitable for recognizing fine gestures. While DeepGRU also used only the coordinates of the hand joints, Devineau *et al.* applied a convolutional operation to fixed-length 1D sequences in the time-domain instead of using a multilayer GRU-RNN. Thus, we assumed that raw 3D data represent complex gestures better than multichannel 1D sequences separately.

Our method slightly outperformed DeepGRU while using fewer parameters; however, DeepGRU has the advantage that the raw data can be used as the input. Additionally, the attention module from DeepGRU improved our method. If DeepGRU uses properly extracted features, its performance could be improved.

Compared to Avola *et al.*'s method, our method showed better performance only for fine gestures. The sampling method proposed by Avola *et al.* only extracted the features from the frames containing abrupt feature value changes and, therefore, redundant features were removed from the input. Although the performance improvement by their sampling method may vary with respect to the selected feature type or the length of the sampled data, the exclusion of inconsiderable frames can be a powerful preprocessing method for dynamic gesture recognition.

Considering the number of parameters, our method requires much less memory than other neural-network-based methods. Our method used approximately 220,000 parameters whereas MC-DCNN used more than 13 million, Avola *et al.* used approximately 1 million, and DeepGRU used more than 3 million.

VI. CONCLUSION AND FUTURE WORK

In this study, we used a PB-GRU-RNN to recognize skeleton-based dynamic hand gestures following noise removal, data normalization, feature part division, and feature extraction. As a result, we obtained better recognition performance than most existing methods. Unlike existing methods that used the entire feature for their input, our method divided the features into multiple parts and used them as inputs for the GRU-RNNs for each hand part. This reduced the number of parameters required for our neural network and improved the performance; therefore, less memory is required to construct HMI systems with neural networks. Furthermore, by properly dividing the feature parts, our method can be modified and used for other skeleton-based dynamic gesture recognition. For other gesture cases, the relationship between the parts and the complexity of the features should be considered to determine the hyperparameters of the GRU-RNNs for each part. Additionally, other than improving neural-network-based classifiers, more accurate preprocessing methods should also be studied to provide better input for classifiers. The proposed method improved recognition performance for dynamic hand gestures by using geometric features; however, erroneous hand skeletons, which may lead to wrong recognition results, need to be handled. We expect that data augmentation methods, such as autoencoders, can serve to resolve this problem. Thus, in future, related issues will be studied to modify the PB-GRU-RNN for joint error suppression and to improve the recognition performance.

REFERENCES

[1] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, Apr. 2011, doi: 10.1145/1922649.1922653.

[2] R. Z. Khan, "Comparative study of hand gesture recognition system," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Technol. Comput. Sci. Inf. Technol. (CS IT)*, Jan. 2012, vol. 2, no. 3, pp. 203–213.

[3] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," *Int. J. Comput. Inf. Eng.*, vol. 3, no. 1, pp. 186–191, Jan. 2009. [Online]. Available: https://publications.waset.org/vol/25

[4] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A hand gesture interface device," in *Proc. SIGCHI/GI Conf. Hum. Factors Comput. Syst. Graph. Interface (CHI)*. New York, NY, USA: Association for Computing Machinery, May 1986, pp. 189–192. [Online]. Available: https://doi.org/10.1145/29933.275628

[5] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015, doi: 10.1007/s10462-012-9356-9.

[6] C.-H. Wu and C. H. Lin, "Depth-based hand gesture recognition for home appliance control," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Hsinchu, Taiwan, Jun. 2013, pp. 279–280.

[7] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Las Vegas, NV, USA, Jun. 2016, pp. 1206–1214.

[8] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 2881–2885.

[9] J. Beh, D. Han, and H. Ko, "Rule-based trajectory segmentation for modeling hand motion trajectory," *Pattern Recognit.*, vol. 47, no. 4, pp. 1586–1601, Apr. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320313004603

[10] H. Cheng, Z. Dai, Z. Liu, and Y. Zhao, "An image-to-class dynamic time warping approach for both 3D static and trajectory hand gesture recognition," *Pattern Recognit.*, vol. 55, pp. 137–147, Jul. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320316000157

[11] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Boston, MA, USA, Jun. 2015, pp. 1–7.

[12] G. Devineau, W. Xi, F. Moutarde, and J. Yang, "Convolutional neural networks for multivariate time series classification using both inter-and intra-channel parallel convolutions," in *Proc. Reconnaissance Formes, Image, Apprentissage Perception (RFIAP)*, Marne la Vallée, France, Jun. 2018, pp. 1–9. [Online]. Available: https://hal-mines-paristech.archives-ouvertes.fr/hal-01888862

[13] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3DCNN and convolutional LSTM," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 1011–1021, Apr. 2019.

[14] M. Maghoumi and J. J. LaViola, Jr., "DeepGRU: Deep gesture recognition utility," in *Advances in Visual Computing*. Cham, Switzerland: Springer, 2019, pp. 16–31.

[15] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Commun. ACM*, vol. 54, no. 2, pp. 60–71, Feb. 2011, doi: 10.1145/1897816.1897838.

[16] H. Zhou, D. J. Lin, and T. S. Huang, "Static hand gesture recognition based on local orientation histogram feature distribution model," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, Washington, DC, USA, Jun. 2004, p. 161.

[17] Y. Fang, K. Wang, J. Cheng, and H. Lu, "A real-time hand gesture recognition method," in *Proc. IEEE Multimedia Expo Int. Conf.*, Beijing, China, Jul. 2007, pp. 995–998.

[18] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proc. 19th ACM Int. Conf. Multimedia (MM)*, New York, NY, USA: Association for Computing Machinery, Nov. 2011, pp. 1093–1096, doi: 10.1145/2072298.2071946.

[19] Y. Li, "Hand gesture recognition using kinect," in *Proc. IEEE Int. Conf. Comput. Sci. Autom. Eng.*, Beijing, China, Jun. 2012, pp. 196–199.

[20] S. P. Priyal and P. K. Bora, "A robust static hand gesture recognition system using geometry based normalizations and krawtchouk moments," *Pattern Recognit.*, vol. 46, no. 8, pp. 2202–2219, Aug. 2013, doi: 10.1016/j.patcog.2013.01.033.

[21] Q. De Smedt, H. Wannous, J.-P. Vandeborre, J. Guerry, B. Le Saux, and D. Filliat, "SHREC'17 track: 3D hand gesture recognition using a depth and skeletal dataset," in *Proc. 10th Eurograph. Workshop 3D Object Retr. (DOR)*, Lyon, France, 2017, pp. 1–6. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01563505

[22] H. Wang, P. Wang, Z. Song, and W. Li, "Large-scale multimodal gesture segmentation and recognition based on convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 3138–3146.

[23] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964.

[24] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 234–245, Jan. 2019.

[25] L. C. Jain and L. R. Medsker, *Recurrent Neural Networks: Design and Applications*, 1st ed. Boca Raton, FL, USA: CRC Press, 1999.

[26] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

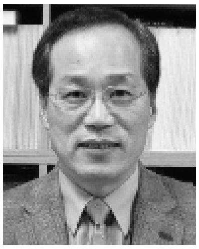
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[28] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 716–723.

[29] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and HOG2 for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Portland, OR, USA, Jun. 2013, pp. 465–470.

[30] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. Del Bimbo, "3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1340–1352, Jul. 2015.

SEUNGHYEOK SHIN was born in Seoul, South Korea, in 1992. He received the B.S. degree in electronic engineering from Hanyang University, Seoul, in 2014, where he is currently pursuing a Ph.D. in Computer Engineering since 2014-2019. His research interest includes gesture recognition and ptychography via neural networks and intelligent vehicles.



WHOI-YUL KIM was born in Bosung, South Korea, in 1956. He received the B.S. degree in electronic engineering from Hanyang University, Seoul, South Korea, in 1980, the M.S. degree in electrical engineering from Pennsylvania State University, PA, USA, in 1983, and the Ph.D. degree in electrical engineering from Purdue University, IN, USA, in 1989. His research interests include computer vision, medical imaging, pattern and object recognition, and intelligent vehicles.