

Research Article

Hand Gesture Classification Based on Nonaudible Sound Using Convolutional Neural Network

Jinhyuck Kim and Sunwoong Choi 

School of Electrical Engineering, Kookmin University, 77 Jeongneung-Ro, Seongbuk-Gu, Seoul, Republic of Korea

Correspondence should be addressed to Sunwoong Choi; schoi@kookmin.ac.kr

Received 31 July 2019; Accepted 25 October 2019; Published 18 November 2019

Academic Editor: Matthew Brodie

Copyright © 2019 Jinhyuck Kim and Sunwoong Choi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recognizing and distinguishing the behavior and gesture of a user has become important owing to an increase in the use of wearable devices, such as a smartwatch. This study is aimed at proposing a method for classifying hand gestures by creating sound in the nonaudible frequency range using a smartphone and reflected signal. The proposed method converts the sound data, which has been reflected and recorded, into an image within a short time using short-time Fourier transform, and the obtained data are applied to a convolutional neural network (CNN) model to classify hand gestures. The results showed classification accuracy for 8 hand gestures with an average of 87.75%. Additionally, it is confirmed that the suggested method has a higher classification accuracy than other machine learning classification algorithms.

1. Introduction

With the advancement in IT technologies, the use of wearable devices, such as a smartwatch or IoT-based devices, is becoming common. However, as devices are getting compact for the ease of convenience, it involves several limitations, such as difficulty in controlling these devices using buttons or touch. Several studies have investigated this issue in the past. It is possible to detect movements of a user using a variety of sensors, such as an infrared light and optical sensor, or to recognize motions using a camera. A device can be controlled based on the recognized movements of a user. In particular, Google Soli [1], which uses radio frequency signals, and Okuli [2], which uses an optic sensor, are aimed at controlling devices through gesture recognition. However, both of these devices require additional parts, such as an optical sensor or a radio frequency chip.

Therefore, methods that recognize gestures without using any additional parts or sensors have attracted a significant amount of attention. The most important method is the one that recognizes gestures using sound wave. In ER [3], which is one of the studies based on gesture recognition using

sound wave, sound waves corresponding to a nonaudible frequency are obtained using a built-in microphone and speaker of a smartphone, and the behaviors are classified based on the Doppler effect. In the abovementioned study, 4 inattentive driving events are classified using a support vector machine (SVM). In SoundWave [4], conducted by Microsoft Research, a laptop with a built-in speaker and microphone was used instead of using a separate converter or a receiver. Nonaudible frequencies are continuously obtained using the built-in speaker and microphone, and gestures are recognized through the reflected signals based on the Doppler effect.

This study proposes the classification of hand gestures, without a separate sensor, using a smartphone. Using smartphone application, we create sound in the nonaudible frequency range and collect sound data reflected from hand gestures. The study extracts the characteristics of the reflected signal based on Doppler effect without using a filter or signal processing. The study distinguishes different hand gestures using a convolutional neural network (CNN) [5], a deep learning model with a high accuracy in image classification. Herein, the reflected and recorded signals are converted

into images and the CNN models are trained. Using the trained model, different hand gestures are classified and the accuracy of classification is evaluated. Resultantly, the study showed an accuracy of 87.75% on 8 hand gestures.

This study is organized as follows. Section 2 introduces the previous studies on gesture recognition using nonaudible frequencies and tracking finger positions. Section 3 discusses the proposed method. Section 4 explains the actual test environment and process and compares the result. Finally, Section 5 describes the conclusion and a direction of future research.

2. Related Work

Various studies that integrated sound wave into IT technology have been conducted. AAMouse [6] suggested a method that controls TV using a smartphone like a remote controller using sound wave generated from a smartphone. Furthermore, there are studies, such as EchoTag [7], that track locations or control devices by combining sound wave with other wireless communication technologies, such as Bluetooth and Wi-Fi. IoT-based devices using sound wave and voice recognition, such as Google Home Mini [8] and Kakao Mini [9], have attracted a significant amount of attention. Therefore, studies based on sound wave data, including sound or voice, are increasing.

In another study, the phase shift of sound wave was used to track finger positions. FingerIO [10] uses an orthogonal frequency division multiplexing (OFDM), which is a modulation technology commonly used by radio communication. A frequency band is divided using OFDM, and the distance is calculated using the actual complex number of each frequency. Strata [11] conducted a study on tracking finger positions using sound wave. In this study, channel impulse response was applied to track a particular channel that corresponds to the figure from the reflected multipath signals. Based on the phase shift of the estimated channel, an absolute distance and a relative distance were calculated to track the position of the finger. In a study that used a low-latency acoustic phase (LLAP) [12], a phase shift of sound wave was converted into the length of an object's movement to track finger positions. A static vector and a dynamic vector were calculated to measure changes and find the position of the finger.

Lastly, there are studies that classify behaviors or gestures using sound wave. ER [3] classified 4 behaviors in a car using a sound wave. It generated sound waves corresponding to a nonaudible frequency of 20 kHz using a smartphone mounted inside of a car and extracted characteristics of each behavior based on the Doppler effect. Then, the behaviors were classified using principal component analysis and SVM, which are one of the machine learning models. Resultantly, it showed a classification accuracy of 94.8% on 4 behaviors. In another study, SoundWave [4], a user's gestures were classified using a laptop. Consecutive pilot tone sounds of 18–19 kHz were obtained using the microphone and speaker built in a laptop, and the reflected signals were analyzed to classify gestures. It classified a total of 5 gestures and showed an accuracy of 96.6%. The abovementioned

study did not use a classifier using separate machine learning or deep learning but mathematically calculated the characteristics of the reflected signals for classification.

The proposed study, unlike other studies using sound wave, obtains data on a specific frequency band for time using short-time Fourier transform [13] and suggests a method for classifying hand gestures with the deep learning model, CNN. In comparison with other studies, this study classified gestures with high mutual similarity and showed a similar level of classification accuracy on more gestures.

3. Our Approach

3.1. System Architecture. We propose a system to classify hand gestures using a smartphone and nonaudible sound. First, we create sound in the nonaudible frequency range and collect sound data reflected from hand gestures using the proposed application. The application comprises a function that generates a nonaudible frequency for a certain period of time and a function that records sound. Two smartphones were used, one being a speaker and another being a microphone. A single-band nonaudible frequency of 20 kHz was generated through the speaker built in a smartphone, and the generated signals were recorded by the microphone built in another smartphone. While the smartphone was recording, we were able to perform each gesture and obtain each different signal reflected accordingly.

Then, the collected data was transferred to a PC and a task to apply to the CNN models was conducted. The recorded data, which is one-dimensional data, were converted into data with the strength per frequency band for time by increasing the data dimension with STFT. During the conversion process, the unwanted frequency band was abandoned and two-dimensional data that extracted a nonaudible frequency were saved.

Finally, the data was learned with the suggested CNN models. The saved data was transferred to a server and divided into learning data and test data to learn with the model and evaluate performance.

3.2. Data Collection. We obtained different signal reflection using hand gesture based on the Doppler effect. The position or movement of a hand for each behavior varied depending on time. Accordingly, reflected and refracted signals were different because of the Doppler effect, causing a difference in the recorded signals. Figure 1 shows sound waves that were recorded differently by behavior due to the Doppler effect. We record and collect data for each behavior repeatedly.

3.3. Data Preprocessing. The recorded sound wave data are converted through STFT to find the frequency characteristics of changing signals over time. Generally, it is possible to obtain changes of signal strength for the overall frequency band over time with STFT. Herein, STFT is applied to the data recorded for 3 sec. The sampling rate of the recorded data is 44.1 kHz. We used the window of size 500 and overlapped 475 size for STFT. The frequency resolution is set as 2048. In the conversion process, the area of interest is cut out by considering the data resolution to only use the

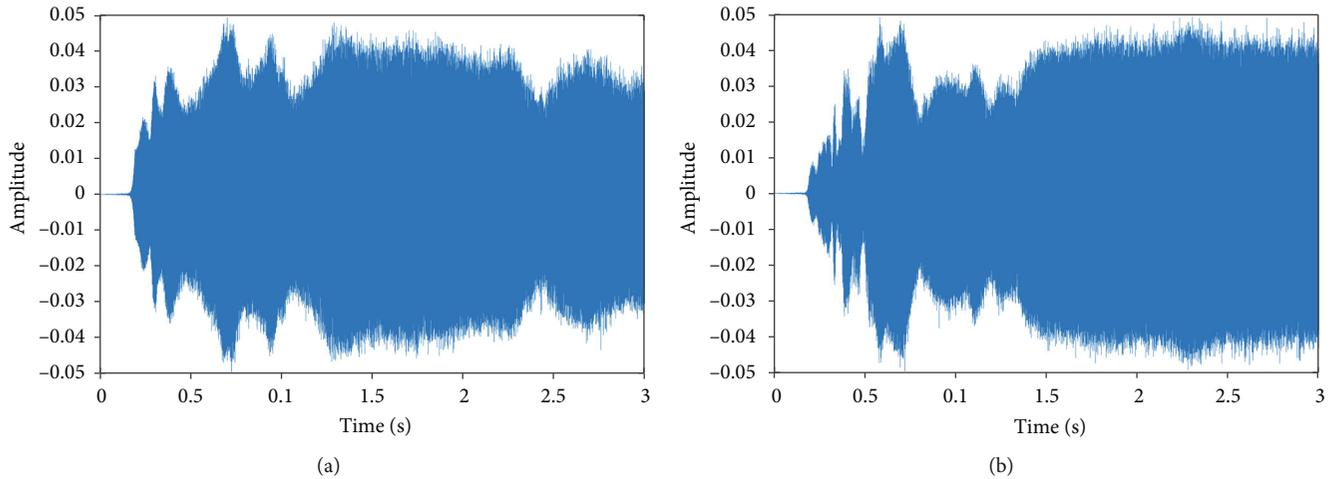


FIGURE 1: A change on sound wave with the Doppler effect. (a) Sound wave recorded when no action is taken. (b) Sound wave recorded when the microphone is blocked.

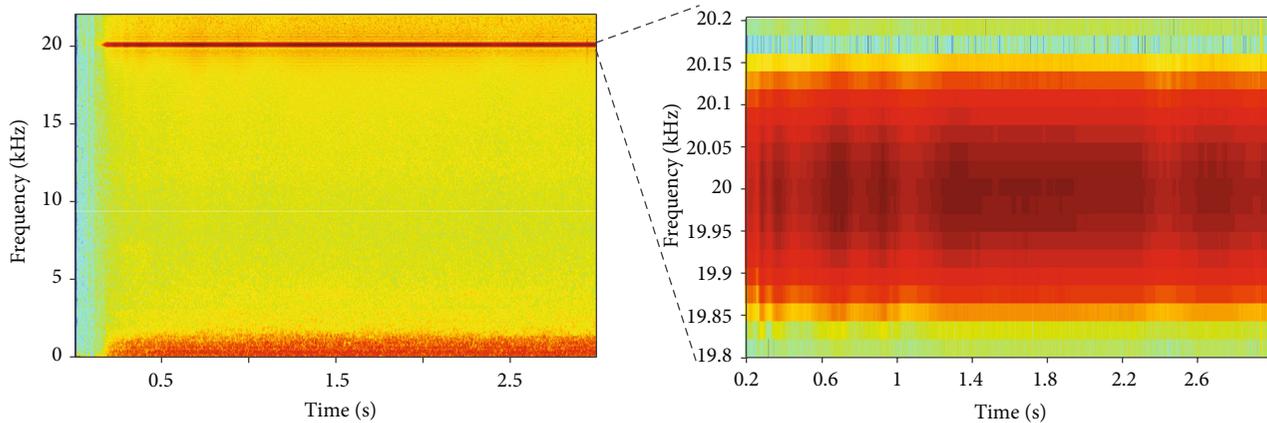


FIGURE 2: STFT result of recorded data and result of cutting out a particular frequency band.

nonaudible frequency band for input data. Figure 2 shows the result of cutting out the 19.8–20.2 kHz section from the STFT result. The color is dark red when the signal in a specific frequency is strong, and the color is blue when the signal is weaker. The starting 0.2 sec is deleted in the cutting process because there is an internal system delay of 0.2 seconds to start recording.

Figure 3 shows that the STFT results differ according to hand gestures. Figure 3(a) is the result of STFT when no action was taken, and it shows different parts with a dark red color compared to Figure 3(b), which is the result of the recorded sound wave when the microphone is blocked.

We obtained the STFT result of the particular section by repeating the above process. The obtained result was saved, and the data were processed to apply to the CNN models.

3.4. Data Learning. For comparison, we made 2 CNN models, a model to learn data before STFT and a model to learn after STFT. The suggested models are composed of 9 layers. Just like general CNN models, they are composed of input, hidden, and output layers. The input layer inputs data

to be classified in compliance with the input format by considering the size of input data. The hidden layer is composed of convolution, pooling, and fully connected layers. The convolution layer is connected to some portion of input data, and it calculates the dot product of the connected domain and its weight. The pooling layer outputs the reduced volume by performing downsampling on dimension. Out of many pooling methods, the suggested models use the max pooling method for selecting the maximum value. In the fully connected layer, all nodes are interconnected and the result of each node is calculated by adding matrix multiplication and deviation of the weight. The suggested models find the average of the corresponding domain through a single average pulling before it is fully connected to reduce the data size. Finally, in the output layer, all classes are converted into probabilities with the Softmax function and they are classified into the class of the highest probability.

Figure 4(a) illustrates the architecture of the CNN model that used raw data of the $1 \times 123480 \times 1$ size as input, which shows the overall architecture of the CNN model that learned raw data among the suggested CNN models. Figure 4(b)

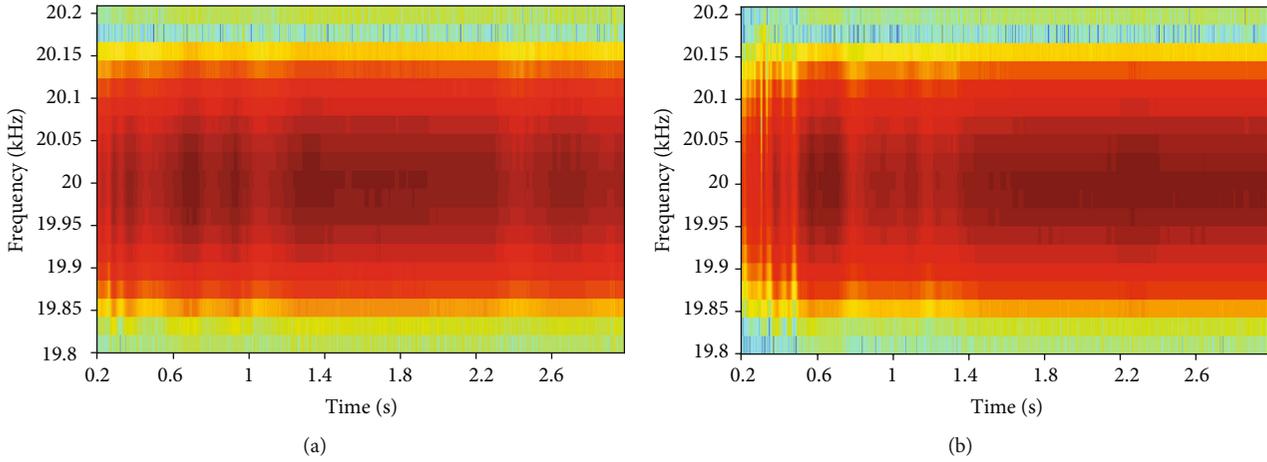


FIGURE 3: Comparison of STFT results on recorded data. (a) STFT result of sound wave recorded when no action is taken. (b) STFT result of sound wave recorded when the microphone is blocked.

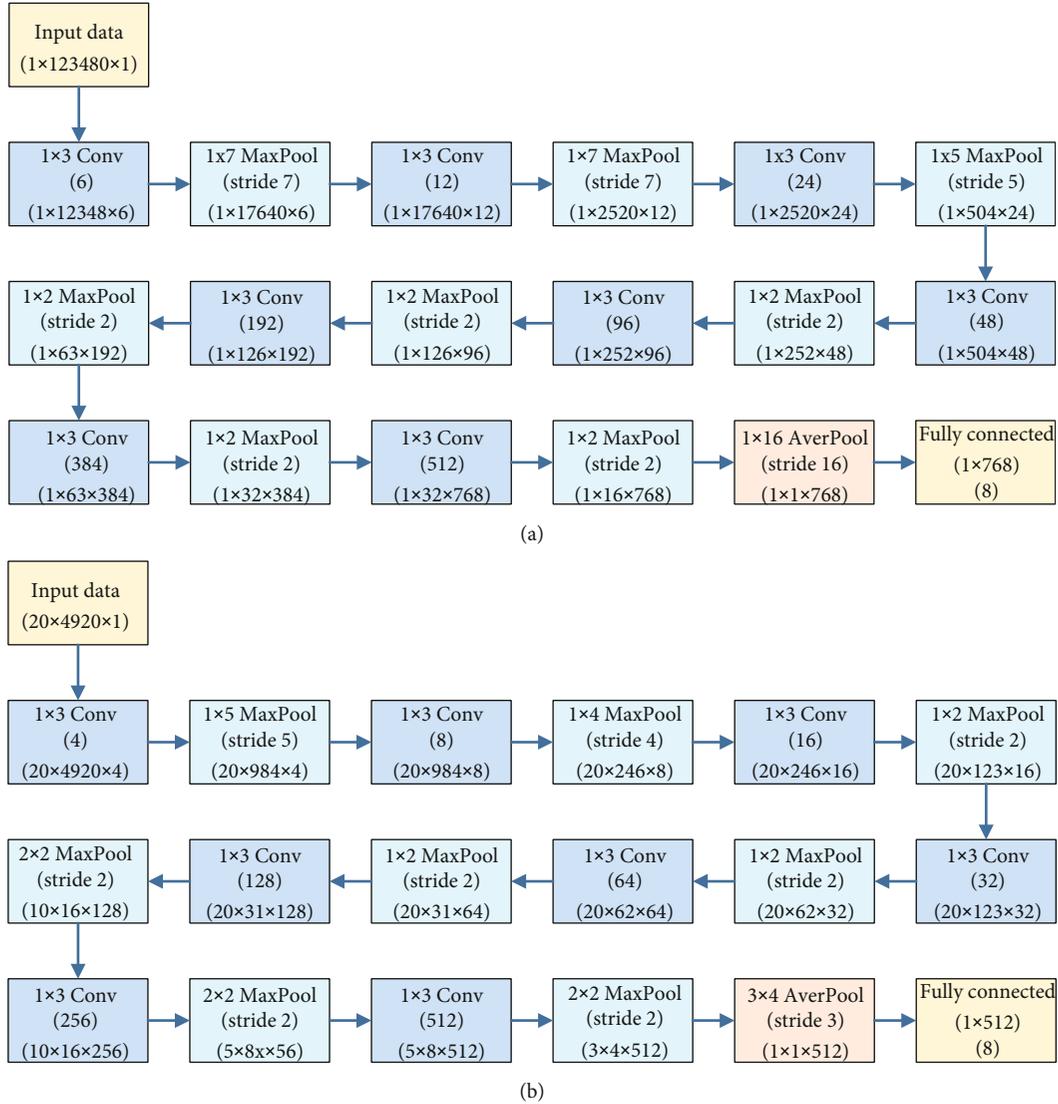


FIGURE 4: Configured CNN models. (a) CNN model applied to raw data of the time domain. (b) CNN model applied to STFT result data.

illustrates the CNN model that used data of the $20 \times 4920 \times 1$ size after STFT as input, which shows the detailed architecture of the CNN model that learned STFT-applied data.

4. Performance Evaluation

4.1. Experiment Setup. Recorded data per direct behavior were collected to evaluate the performance of the proposed method. For data collection, we made an application that generates and records a single nonaudible frequency of 20 kHz. We used 2 smartphones for this experiment. Samsung Galaxy S8 model was used as a speaker that generates nonaudible frequencies, and Samsung Galaxy Note 8 model was used as a microphone for recording sounds. The application was installed on each smartphone to proceed with the experiment. The experiment was performed in a deserted laboratory. Two smartphones were placed on a table with space, and the PLAY button is pushed on the smartphone that works as a speaker. Then, the REC-START button is pushed on the smartphone that works as a microphone before taking particular hand gestures. Data is collected by repeating this process.

After data collection is completed, the recorded data saved in the smartphone are transferred to PC for STFT. We used MATLAB R2015 for data conversion. After applying STFT using MATLAB, the result data are saved as a file. Then, the saved data of 2 groups were transferred to the server to apply to the CNN models.

A GPU server was used to implement and test the CNN models. The GPU server used the GTX 1080 Ti model. With TensorFlow, we implemented 2 CNN models to learn and evaluate data before and after applying STFT.

4.2. Gesture Dataset. This study classifies 8 hand gestures as follows:

- (1) Do nothing: do nothing while recording
- (2) Move from left to right: move the hand from left to right with unfolded palm while recording. At this time, the hand is positioned over the screen of the recording smartphone
- (3) Move from top to bottom: move the hand from top to bottom with unfolded palm while recording. At this time, the hand is positioned over the screen of the recording smartphone
- (4) Circle drawing: draw a circle clockwise with the unfolded index finger while recording. At this time, the finger is positioned over the screen of the recording smartphone
- (5) Block the microphone: block the microphone on the bottom of the smartphone with the palm while recording
- (6) Move from right to left: move the hand from right to left with unfolded palm while recording. At this time, the hand is positioned over the screen of the recording smartphone

- (7) Move from bottom to top: move the hand from bottom to top with unfolded palm while recording. At this time, the hand is positioned over the screen of the recording smartphone
- (8) Triangle drawing: draw a triangle clockwise with the unfolded index finger while recording. At this time, the finger is positioned over the screen of the recording smartphone

Figure 5 illustrates hand gestures collected for the experiment. Each hand gesture was completed within 3 seconds. We collected data by repeating 8 hand gestures as illustrated. The hand did not touch the smartphone screen while making hand gestures and was positioned about 1 cm away from the screen.

Data on total 8 types of hand gestures were collected and used for the experiment. Each data was recorded for 3 sec, and a total of 800 data, 100 times per hand gesture, were collected. Data collection was performed in a noise-free laboratory.

4.2.1. Noisy Environment. For comparison, data were obtained in a noisy environment. We created an environment with noise in the same laboratory with a laptop. To collect data, music was played using the built-in speaker of the laptop while recording. Data with noise were recorded for 3 sec for each of 8 gestures using the same method, and a total of 800 data, 100 times per gesture, were collected.

As a result, 100 data under a noise-free environment and 100 data under a noisy environment were collected for each hand gesture. Table 1 shows the recording data used.

After collecting 1600 data samples, STFT was applied using MATLAB. Then, the 19.8–20.2 kHz domain, which is the nonaudible frequency band, was cut out. We let each CNN model learn data that applied STFT and data that did not apply STFT and evaluated the performance. For each group, the tests were conducted on the scenarios with and without noise, respectively. Learning data and evaluation data were divided into 8:2. To decide an appropriate repetition count of learning, 1 epoch was set to 400 to check accuracy. When the total number was set to 20000, the test result was obtained as in Figure 6. Based on this result, we set the repetition count of learning to 37.5 epoch and proceeded with evaluation.

4.3. Gesture Dataset

4.3.1. Effect of STFT. For comparison, data before STFT was applied to the CNN models to learn and evaluate performance. The K -fold cross-validation method [14] was used to increase the reliability of the classification result. In this experiment, K was set to 5 to perform learning and evaluation. As a result of the evaluation, maximum and average classification accuracies were 81.25% and 76.25%, respectively. Figure 7 shows the average of 5 K -fold cross-validation results as a confusion matrix.

Then, data after STFT was applied to CNN to learn and evaluate performance. Average accuracy was 87.85%, which is over 10% higher than the result before STFT. Maximum

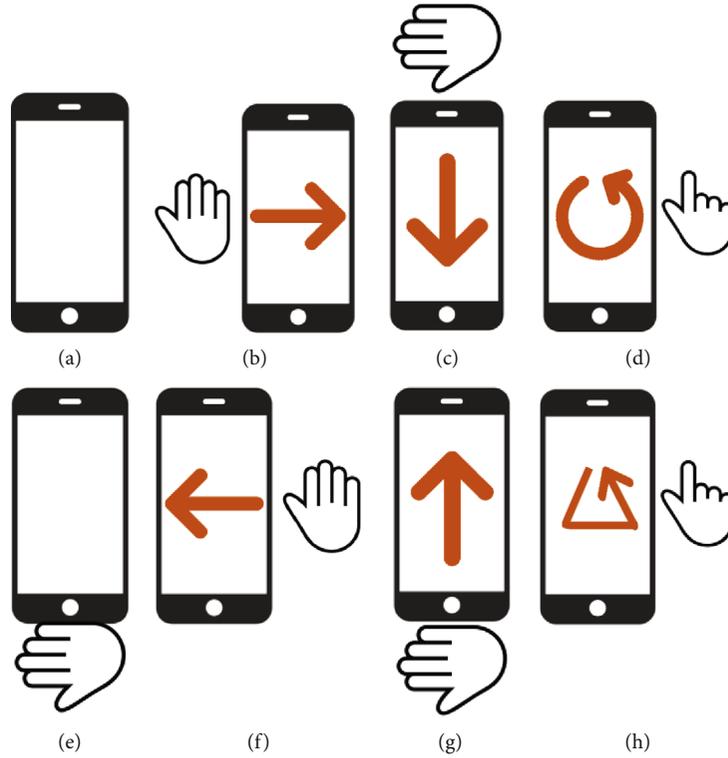


FIGURE 5: Used gestures. (a) Do nothing. (b) Move from left to right. (c) Move from top to bottom. (d) Circle drawing. (e) Block the microphone. (f) Move from right to left. (g) Move from bottom to top. (h) Triangle drawing.

TABLE 1: Recording data used.

| Hand gesture | Number of general data | Number of data with noise |
|------------------|------------------------|---------------------------|
| Do nothing | 100 | 100 |
| Left to right | 100 | 100 |
| Top to bottom | 100 | 100 |
| Drawing circle | 100 | 100 |
| Block the mic | 100 | 100 |
| Right to left | 100 | 100 |
| Bottom to top | 100 | 100 |
| Drawing triangle | 100 | 100 |

classification accuracy was 92.5%. Figure 8 shows the evaluation result as a confusion matrix.

Figure 9 compares the results of 2 models with graphs. The differences in accuracy between STFT-applied data and data without STFT were 11.25% and 15% for maximum and minimum accuracies, respectively. When average accuracy is compared, STFT-applied data showed 11.5% higher classification accuracy.

4.3.2. Effect of Noise. Subsequently, we added data with noise to evaluate performance. Before comparing the effect of noise, the same process was conducted to validate whether application of STFT is effective. Learning and evaluation of 2 models were performed using K -fold cross-validation, and the results were compared.

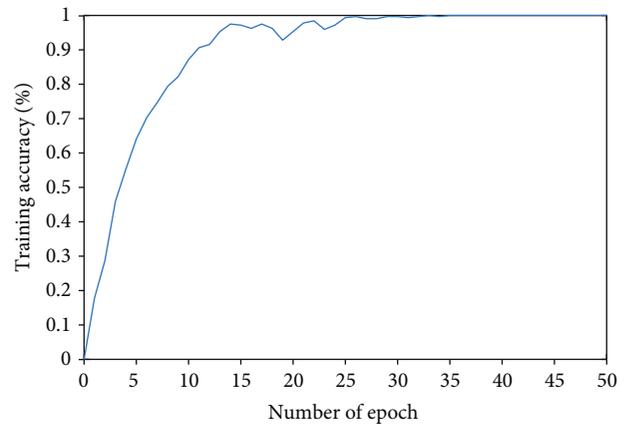


FIGURE 6: Accuracy based on number of training per epoch.

When data without STFT were used, an average classification accuracy of 74.18% was obtained. Maximum accuracy was 78.13%, which is relatively lower than data without noise. Figure 10 shows the evaluation result as a confusion matrix.

When the STFT-applied data were used, an average classification accuracy of 79.38% and maximum accuracy of 88.48% were obtained, respectively. Likewise, it showed relatively lower accuracy than data without noise. When compared to data that did not apply STFT, it showed over 5% higher average accuracy. Figure 11 shows the result of adding data with noise and applying STFT for classification as a confusion matrix.

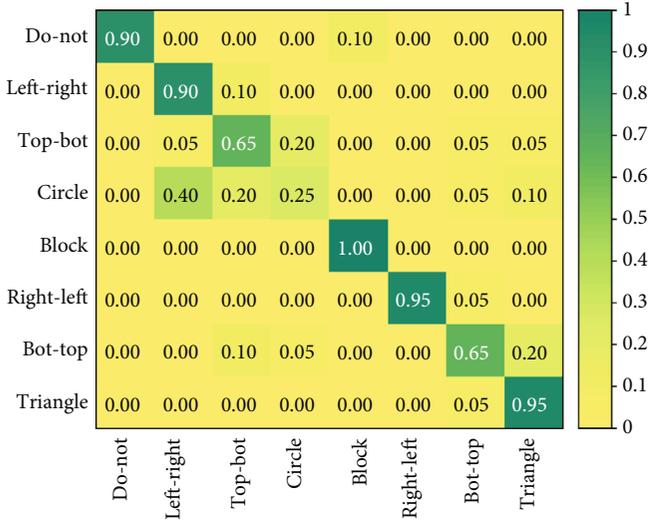


FIGURE 7: Confusion matrix of the result that applied raw data to the CNN model.

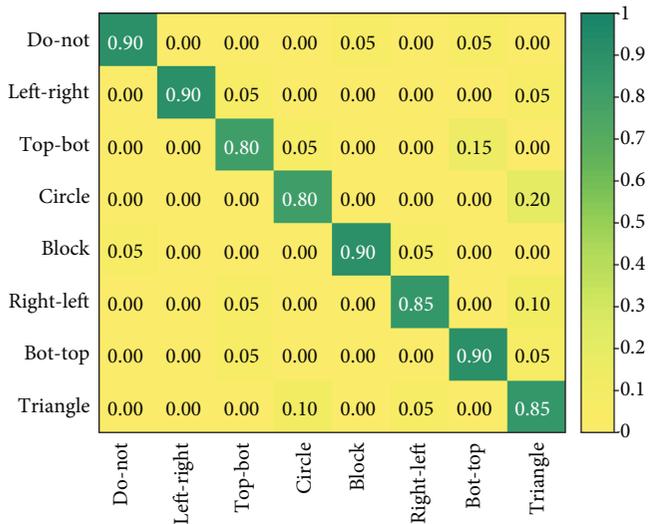


FIGURE 8: Confusion matrix of the result that applied STFT data to the CNN model.

Data with noise also showed higher accuracy when STFT was applied. When data with noise were used, the model that applied STFT showed over 15% higher average accuracy than the model that did not apply STFT. Then, we compared the effect of noise. When there was noise in STFT-applied data, it showed about 5% lower average accuracy than data without noise, and data that did not apply STFT but has noise showed 11.5% lower accuracy than data without noise. Figure 12 compares the classification accuracy of STFT-applied data with and without noise using graphs.

Table 2 shows the overall experiment result. For the model that used STFT-applied data and the model that used raw data, accuracy of hand gesture classification was compared in the environment with noise and environment with-

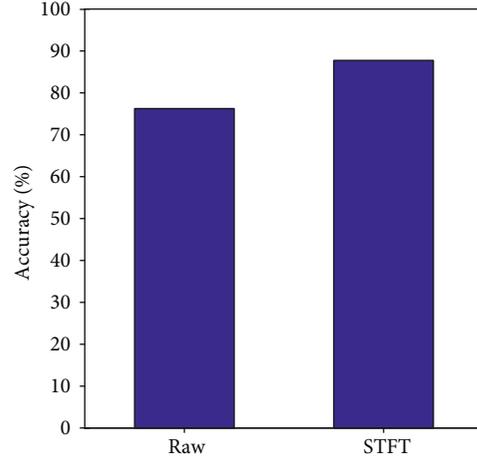


FIGURE 9: Comparison of classification accuracy of CNN models using raw data and STFT-applied data.



FIGURE 10: Confusion matrix of data that applied raw data to the CNN model with noise data.

out noise. Average, maximum, and minimum accuracies are compared for the result of K -fold cross-validation.

4.3.3. Comparison with Other Classification Algorithms. Additionally, we compared the result using other machine learning methods used for data classification. For input data, STFT-applied data were used. Machine learning methods used for comparison are decision tree (DT), SVM, and random forest (RF).

DT [15] is an estimation method for associating the observed value and the target value of a specific item. It analyzes data and shows the pattern existing between these values as a combination of predictable rules. The max depth of the DT is set to 4 in this study.

SVM [16] finds a boundary that creates the biggest margin between 2 data. It can process complex data effectively

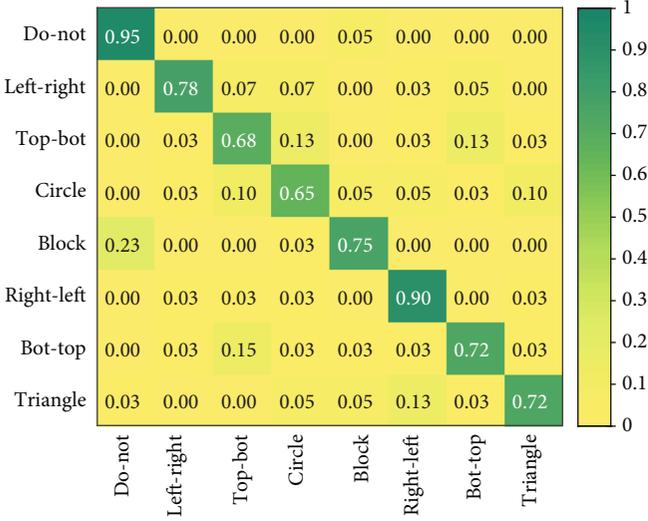


FIGURE 11: Confusion matrix of data that applied STFT to the CNN model with noise data.

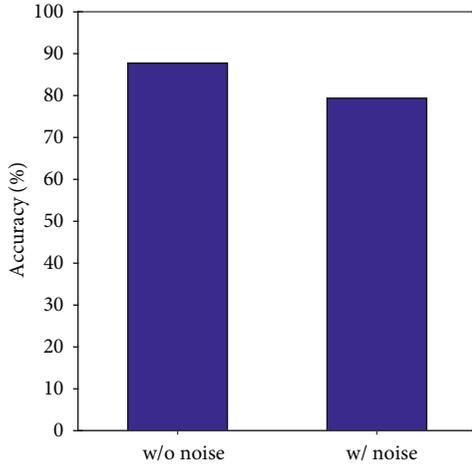


FIGURE 12: Comparison of classification accuracy of CNN models using STFT-applied data with and without noise.

TABLE 2: Average, minimum, and maximum classification accuracies according to data and environment.

| Environment | Input data | Accuracy (%) | | |
|-------------|------------|--------------|---------|---------|
| | | Average | Minimum | Maximum |
| w/o noise | Raw | 76.25 | 70.62 | 81.25 |
| | STFT | 87.75 | 85.62 | 92.5 |
| w/ noise | Raw | 74.18 | 69.37 | 78.12 |
| | STFT | 79.37 | 72.18 | 84.37 |

using Kernel. We set the normalization parameter (C) between 1 and 10000 as 100 and the gamma parameter (γ) between 0.0001 and 1 as auto.

RF [17] is an ensemble classification method that randomly learns multiple DTs. It creates a number of decision trees, determines the result by majority vote, and predicts the result. We evaluated RF performance using 200 trees.

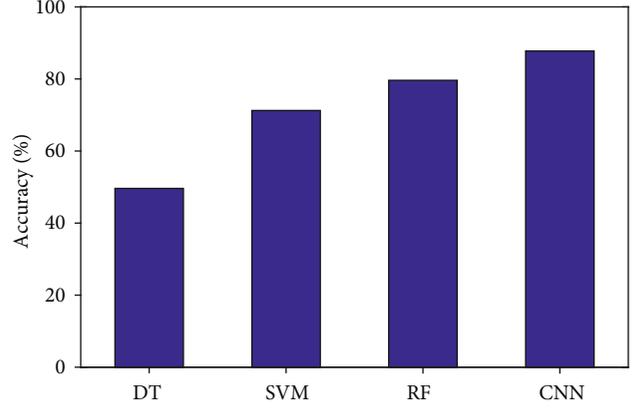


FIGURE 13: Comparison of accuracy on 8 behaviors using different algorithms.

Classification algorithms are implemented on the server for evaluation. As a result, classification accuracy on 8 behaviors is 49.63%, 71.25%, 79.63%, and 87.75% in order. Resultantly, the suggested method showed better performance than existing classification algorithms. Figure 13 compares the accuracy of 3 machine learning algorithms and the suggested CNN models.

5. Conclusions

This study classified hand gestures using nonaudible frequencies of a smartphone. Herein, a method was presented for finding the frequency responses over time for data classification using STFT and applying them to the CNN models. The signals recorded as one-dimensional data were converted into two-dimensional data, and the CNN model with high accuracy in image classification was used to find characteristics that cannot be obtained from one-dimensional data and increase the accuracy of classification. The suggested method showed better performance than the existing machine learning classification models with an accuracy of 87.75% accuracy regarding 8 hand gestures. An average classification accuracy of 79.38% was obtained when the data with noise were collected. In comparison to classification using the CNN model without applying STFT, it showed over 10% higher accuracy without noise and over 5% higher accuracy without noise.

In the future, an additional test will be conducted by considering other gestures and changing the place and environment. A study on applying a method to handle unlearned hand gestures instead of setting gestures in advance will be conducted. If a system is constructed by supplementing and storing the learning model, it is expected that it will be possible to handle hand gestures in real time.

Data Availability

The raw data is available at “<http://ncl.kookmin.ac.kr/data/sound2019.zip>” or from the corresponding author upon request.

Disclosure

The abstract of the manuscript is presented as Conference in 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) (No. 2016R1A5A1012966). The authors gratefully acknowledge Mr. Jinwon Cheon for the help in collecting data.

References

- [1] J. Lien, N. Gillian, M. E. Karagozler et al., "Soli: ubiquitous gesture sensing with millimeter wave radar," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–19, 2016.
- [2] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, "Extending mobile interaction through near-field visible light sensing," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pp. 345–357, Paris, France, September 2015.
- [3] H. Gao, X. Xu, J. Yu et al., "ER: early recognition of inattentive driving leveraging audio devices on smartphones," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Atlanta, GA, USA, May 2017.
- [4] S. Gupta, D. Morris, S. N. Patel, and D. Tan, "SoundWave: using the Doppler effect to sense gestures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1911–1914, Austin, Texas, USA, May 2012.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [6] S. Yun, Y. C. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proceedings of the 13th annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 15–29, Florence, Italy, May 2015.
- [7] Y. C. Tung and K. G. Shin, "EchoTag: accurate infrastructure-free indoor location tagging with smartphones," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 525–536, Paris, France, September 2015.
- [8] Google Home MiniOctober 2018, http://www.store.google.com/product/google_home_mini.
- [9] K. MiniOctober 2018, <http://www.kakao.ai/product/kakaomini>.
- [10] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "FingerIO: using active sonar for fine-grained finger tracking," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pp. 1515–1525, San Jose, California, USA, May 2016.
- [11] S. Yun, Y. C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: fine-grained acoustic-based device-free tracking," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 15–28, Niagara Falls, New York, USA, June 2017.
- [12] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 82–94, New York, USA, October 2016.
- [13] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.
- [14] T. Fushiki, "Estimation of prediction error by using K-fold cross-validation," *Statistics and Computing*, vol. 21, no. 2, pp. 137–146, 2011.
- [15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [16] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer Science & Business Media, Berlin, Germany, 2008.
- [17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.



Hindawi

Submit your manuscripts at
www.hindawi.com

