IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 25, 2021, accepted April 12, 2021, date of publication May 10, 2021, date of current version June 9, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3078715*

# Wav2KWS: Transfer Learning From Speech Representations for Keyword Spotting

**DEOKJIN SEO[1], HEUNG-SEON OH[2], AND YUCHUL JUNG[1]**
[1]Department of Computer Engineering, Kumoh National Institute of Technology (KIT), Gumi 39177, South Korea
[2]School of Computer Science and Engineering, Korea University of Technology and Education (KOREATECH), Cheonan 31253, South Korea

Corresponding author: Yuchul Jung (jyc@kumoh.ac.kr)

This work was supported in part by the Ministry of Trade, Industry and Energy (MOTIE), in part by the Korea Institute for Advancement of Technology (KIAT) through the National Innovation Cluster Research and Development Program under Grant P0006704, in part by the National Research Foundation of Korea (NRF) by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant NRF-2018R1C1B5031408, in part by the Education and Research Promotion Program of Korea University of Technology and Education (KOREATECH), in 2021, and in part by the High Performance Computing (HPC) Support Project by the MSIT and the National IT Industry Promotion Agency (NIPA).

**ABSTRACT** With the expanding development of on-device artificial intelligence, voice-enabled devices such as smart speakers, wearables, and other on-device or edge processing systems have been proposed. However, building or obtaining large training datasets that are essential for robust keyword spotting (KWS) remains cumbersome. To address this problem, we propose a deep neural network that can rapidly establish a high-performance KWS system from arbitrary keyword instruction sets. We use an encoder pretrained with a large-scale speech corpus as the backbone network and then design an effective transfer network for KWS. To demonstrate the feasibility of the proposed network, various experiments were conducted on Google Speech Command Datasets V1 and V2. In addition, to verify the applicability of the network for different languages, we conducted experiments using three different Korean speech command datasets. The proposed network outperforms state-of-the-art deep neural networks in both experiments. Furthermore, the proposed network can understand real human voice even when trained with synthetic text-to-speech data.

**INDEX TERMS** Keyword spotting, speech commands recognition, transfer learning.

## I. INTRODUCTION

Deep learning has enabled the application of automatic speech recognition (ASR) to commercial services [1]–[7]. However, several computational resources are required to create an ASR model based on neural networks, and powerful graphics processing units may be unavailable for some applications. Moreover, ASR and related tasks may lack sufficient training data. Hence, pretrained models provide a solution to overcome data scarcity in the target domain. Recently, Schneider *et al.* [5] showed that a pretrained model with LibriSpeech [8] can obtain high performance by fine-tuning a model with a small training dataset. Similarly, bidirectional encoder representations from transformers (BERT) pretraining has been successfully applied to natural language processing [9]. Nevertheless, it remains difficult to obtain data suitable for various target domains.

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

Keyword spotting (KWS) recognizes utterances of a limited number of commands and is being actively studied along with ASR that recognizes continuous speech [10]–[14]. The constrained commands in KWS include simple control keywords, such as yes or no. Popular commands include "Hey Siri" [15], "Alexa" [16], [17], and "Okay Google" [10], which trigger the control of devices from specific vendors. Commands to control applications and services include "play the music," "turn off," and "how is the weather tomorrow?" While the applicability of neural networks to KWS has been demonstrated, recent studies have pursued performance improvement and reduction in the number of parameters [18]–[20], and other studies have focused on improving the real-time KWS performance [12], [21].

The Google Speech Command Datasets [22] are considered as the de facto KWS standard. Unfortunately, KWS has considerably lesser publicly available data than ASR, hindering the composition of arbitrary keywords and the construction of comprehensive KWS systems during data preparation. Consequently, neural network optimization becomes difficult

given the scarce training data available. To overcome data scarcity for KWS, a pretrained head model and synthesized data have been used [23]. Lin *et al.* [23] state that building a state-of-the-art (SOTA) KWS model requires more than 4000 recorded human speech samples per command.

In this study, we devised a method to maximize the recognition of speech command utterances even if few command samples are available by adopting the Wav2Vec 2.0 framework [5], which is detailed in Section III-A. To achieve a high KWS performance, we adopt a suitable encoder for learning transferred speech representations from a backbone [24], as shown in Fig. 1.

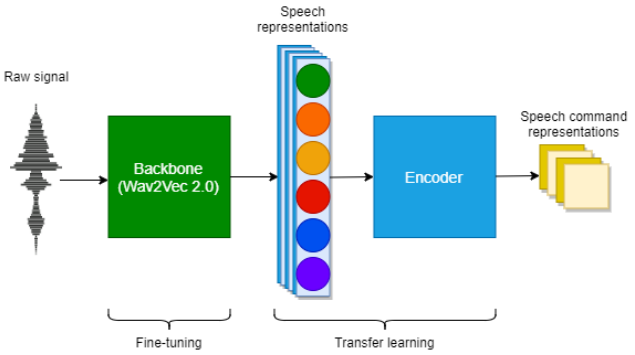



**FIGURE 1. The backbone uses a pretrained encoder of Wav2Vec 2.0, and an additional encoder is combined for KWS. The backbone is fine-tuned, and the encoder is trained with speech command representations transferred from speech representations.**

To describe KWS as a classification problem, we consider that the training samples in the backbone are longer than the short utterances of speech commands. Then, we design an encoder that enables transfer learning [25]. During fine-tuning, the backbone and encoder perform optimization on the target KWS domain dataset.

The proposed model applied to speech command recognition improves the SOTA performance on Google Speech Command Datasets V1 [26] and V2 [27]. Moreover, the proposed model achieves more than 99% accuracy in the Korean speech command datasets that we constructed. To the best of our knowledge, this is the first attempt to show that a model pretrained on an English speech corpus can be successfully applied to KWS using Korean speech data. We test our model on three different types of Korean commands with terminology related solar power systems, healthcare, and daily activities. Each dataset contains less than 300 samples per command.

The contributions of this study can be summarized as follows:

1) Using a pretrained model based on a large ASR corpus as backbone, we propose a model to transfer a vector of wav representations to KWS representations (Wav2KWS).
2) The proposed Wav2KWS model achieves a SOTA performance, being superior to existing models on Google Speech Command Datasets V1 and V2 and three Korean speech command datasets.
3) We show that high-performance KWS can be achieved with a small speech command dataset by using text-to-speech (TTS) synthesized data.
4) We also show that a backbone pretrained on an English speech corpus can be used for KWS in other languages such as Korean by applying transfer learning.

## II. RELATED WORK

### A. ASR

ASR is being rapidly developed, and emerging models keep increasing the SOTA performance. Recently, Wav2Vec 2.0 [5] achieved SOTA performance in ASR, being comparable to BERT [9] in natural language processing. Wav2Vec 2.0 is pretrained on the LibriSpeech Corpus [8]. In natural language processing, BERT is pretrained using masked language modeling [28], [29] and then fine-tuned for various downstream tasks [30]–[32]. On the other hand, Wav2Vec 2.0 relies on end-to-end fine-tuning [5], [6], [33].

### B. KWS

With the advent of edge computing, research on KWS based on deep learning has been devoted to increase the performance by achieving a faster inference or decreasing the number of parameters [19], [20]. Temporal convolution [19] can reduce the number of parameters of existing models, and a 1D time-channel separable convolutional neural network [20] has further lightened the model. Likewise, the performance degradation after reducing the size of a ResNet model has been prevented by using data augmentation [34].

Recent studies on KWS have considered various problems, such as recognition in real-world environments and robustness to noise besides the reduction in the number of parameters. A multi-head attention recurrent neural network has achieved a high accuracy of 97.2% and 98.0% on Google Speech Command Datasets V1 and V2 [21], respectively, thus overcoming the gap in model performance between streaming KWS and test datasets. Although properly training a neural network depends on the availability of training data, solving data scarcity may be cumbersome and costly.

A head model has shown the benefits of embedding, which is built on learning many short utterances [23]. The head model quickly converges to the model from the shared weights of pretrained embeddings. Without the head model, 4000 real samples are required for training, whereas only 500 samples are required to achieve 97.7% accuracy on Google Speech Command Dataset V2 when using the head model. Although improved end-to-end models [21] (e.g., QuartzNet [35]) for KWS are available, a pretrained model on a large-scale speech corpus has not been used for KWS.

We achieve accurate KWS with a small training dataset even if there are not additional synthesized data. By using a pretrained model as backbone, we propose a simple

yet effective structure to efficiently learn speech command representations.

## III. PROPOSED KWS MODEL

### A. WAV2VEC 2.0 ENCODER

The Wav2Vec 2.0 encoder is trained by contrastive loss (1) that decreases the cosine similarity with $c_t$ between $q_t$ and $\tilde{q}$ for identifying the true quantized latent speech representation, $q_t$, in a set of $K+1$ quantized candidate representations $\tilde{q} \in Q_t$. The features are extracted from the input raw audio signal by a convolutional layer with a wide kernel as feature extractor and then transformed into high-level features through sequential convolutional layers and post-projection. The encoded features are combined with relative positional embeddings calculated by group convolutional layers before being input to a transformer. The output of each convolutional layer goes through gaussian error linear unit (GeLU) [36] activation to increase nonlinearity and perform layer normalization. GeLU matches or exceeds the performance of models with rectified linear unit (ReLU) or exponential linear unit activation in various tasks including ASR. Then, by capturing the speech representations from each transformer layer at $t$ time steps, $C_T$: $c_1, \ldots, c_t$ is obtained as the transformer output and connected to a new encoding stage in the transfer learning network.

$$contrastive\ loss = -log \frac{exp\left(sim\left(c_t, q_t\right)/k\right)}{\sum_{\tilde{q}\sim Q_t} exp\left(sim\left(c_t, \tilde{q}\right)/k\right)} \quad (1)$$

As the abovementioned process depends on the quantization codebook [37], it has been designed to leverage the available codebook and use diversity loss. The weight of the diversity loss is adjusted by its hyperparameters.

### B. TRANSFER LEARNING NETWORK

Fig. 2 shows a diagram of the proposed Wav2KWS model, in which a backbone network and a transfer learning network are connected. The backbone network based on Wav2Vec 2.0 is pretrained on LibriSpeech 960 hours. The transfer learning network includes a novel encoder that enables transfer learning of encoded speech representations into speech command features.

In the transfer learning network, we consider the following two aspects and implement the necessary layers: 1) the data used for pretraining of the backbone network differ from speech commands data. Thus, additional encoding is required for representing specific data characteristics as vectors; 2) contextualized representations of backbone output are vectors with time series; thus, they should be represented as a set of speech commands while shrinking the time series for classification.

To address these two aspects, we first condense time series information by transforming the transformer output ($C_T$) into $C$ while maintaining context information from a convolutional layer with a kernel size adapted to the speech sampling rate in a time series vector. Then, we compress information using a pointwise convolutional layer and finally

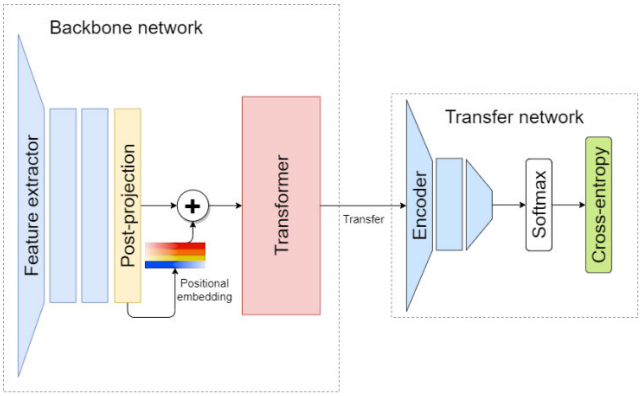



**FIGURE 2.** Structure of proposed Wav2KWS model. A backbone network is directly connected to the transfer network. The backbone network comprises a convolutional layer, feature extraction, post-projection, positional embedding, and a transformer. The transfer network consists of convolutional layers of receptive field, pointwise and 1D convolutional layers for a sequential output. The detailed hyperparameters of the Wav2KWS model are described in Fig. 3 and Table 1.

use a 1D convolutional layer as the classifier to preserve nonlinear information from both the backbone network and the transfer learning network. The outputs of the first two convolutional layers go through ReLU activation to increase nonlinearity and perform layer normalization. The Wav2KWS model uses a softmax function to solve $L_c$ in (2) by classifying $k$ speech command sets $\mathrm{X}(x_0, \ldots, x_k)$ after training the latent speech command representations in the transfer learning network.

$$L_c = -log\left(\frac{exp(x)}{\sum_j exp\left(C_j\right)}\right) \quad (2)$$

Fig. 3 and Table 1 detail the proposed Wav2KWS model. The output of each convolutional layer uses GeLU activation in the backbone and ReLU activation in the transfer learning network, and all the outputs undergo layer normalization. The inset on the right of Fig. 3 shows the conversion of contextualized speech representations encoded in the backbone into speech command representations.

The first feature extractor consists of one convolutional layer and one block. In the first convolutional layer, a kernel of size 10 extracts features from the raw audio signal. The parameters of Block A in Fig. 3 are listed in Table 1. Layer Conv2 is repeated three times. Then, layer Conv3 is repeated twice. Positional embedding uses a kernel of size 128 and 16 groups of convolutional layers to encode relative positional information and add features.

The transformer has 768 hidden dimensions and provides contextualized speech representations to the transfer learning network. Then, two convolutional layers, Conv4 and Conv5, constitute a novel encoder for transfer learning. Layer Conv4 uses a kernel of size 25 to compress the speech time series. Layer Conv5 uses pointwise convolution. The last layer, Conv6, is used for classification, and a 1D convolutional layer preserves nonlinear and location information.

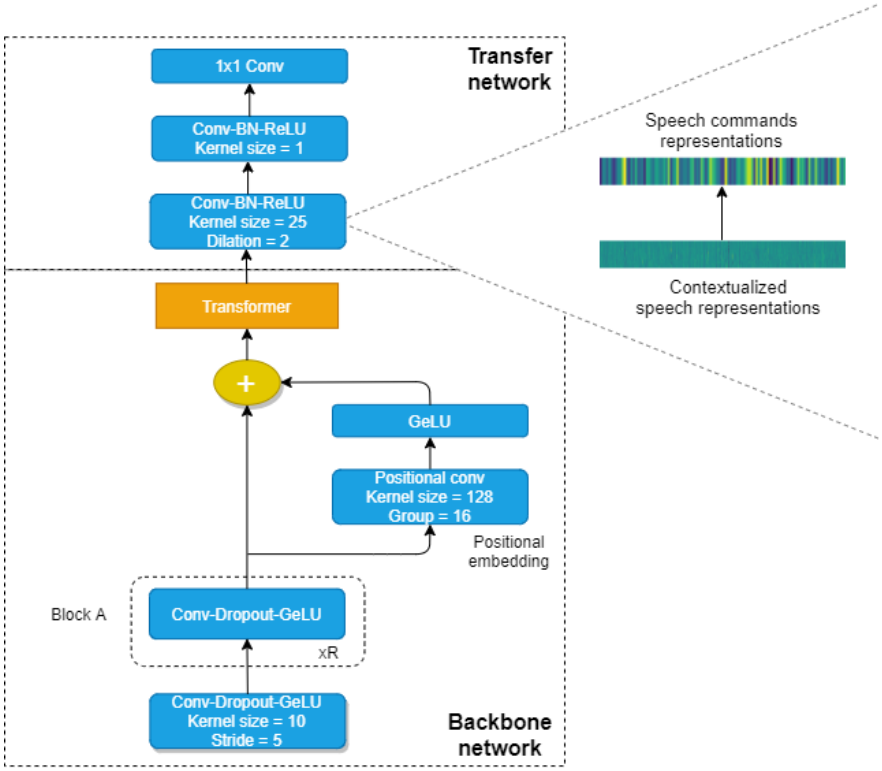



**FIGURE 3.** Architecture of proposed Wav2KWS model. (Omitted hyperparameters use their default values. R, number of recurrent layers; Conv, convolutional layer; BN, batch normalization; ReLU, rectified linear unit; GeLU, gaussian error linear unit).

**TABLE 1.** Hyperparameters of proposed Wav2KWS model. (K, kernel size; S/D/G, stride, dilation, and groups; R, number of recurrent layers).

| Block | No. output channels | K | S/D/G | Dropout | R |
|---|---|---|---|---|---|
| Conv1 | 512 | 10 | 5/1/1 | 0.1 | 1 |
| Block A, Conv2 | 512 | 3 | 2/1/1 | 0.1 | 3 |
| Block A, Conv3 | 512 | 2 | 2/1/1 | 0.1 | 2 |
| Positional Conv | 768 | 128 | 1/2/16 | – | 1 |
| Conv4 | 112 | 25 | 1/2/1 | – | 1 |
| Conv5 | 112 | 1 | 1/1/1 | – | 1 |
| Conv6 | No. classes | 1 | 1/1/1 | – | 1 |
| Softmax | – | – | – | – | – |
| Cross-entropy | – | – | – | – | – |

This final layer has as several channels as the number of classes.

## IV. EXPERIMENTS

We used the "Wav2Vec 2.0 Base (No fine-tuning)" pretrained models of Wav2Vec 2.0 (available at https://github.com/pytorch/fairseq/tree/master/examples/wav2vec) [5] as the backbone of the proposed Wav2KWS model. There are several large models, but they have many parameters for learning representations of limited speech commands. When input data were processed, the raw signal was separated into 1 s windows without extracting the spectrograms. Then, we applied three data augmentation operations (Section IV-A). We chose Adam optimization [38] for training and set the learning rate of the backbone to 1e−5 for fine-tuning. In addition, the learning rate of each newly added convolutional layer was set to 5e–4 for speech command learning, and an L2 penalty of 1e–5 was established.

### A. DATA AUGMENTATION AND PREPROCESSING

The sampling rate of each speech sample was converted to 16 kHz. To extract a representative 1 s segment with the largest rms value from each speech sample, we implemented the loudest section extraction [39] in Python (Fig. 4). To this end, 1 s windows with 1 ms overlap were analyzed.

During training, the following three data augmentation operations were applied:

1) Time shifting the signal by time $t_s$ selected from a uniform distribution in [−100, 100) ms.
2) Background noise synthesized with power $p$ selected from a uniform distribution in [0, 0.7) [12], [40].
3) Similar to SpecAugment(…)[41], silence masking of signals by setting the signal value to 0 from time $m$ until $m+$ 100 ms according to a uniform distribution in [−100, 100) ms (Fig. 5).

We applied every data augmentation operation with a probability of 0.5, achieving the highest performance when all the

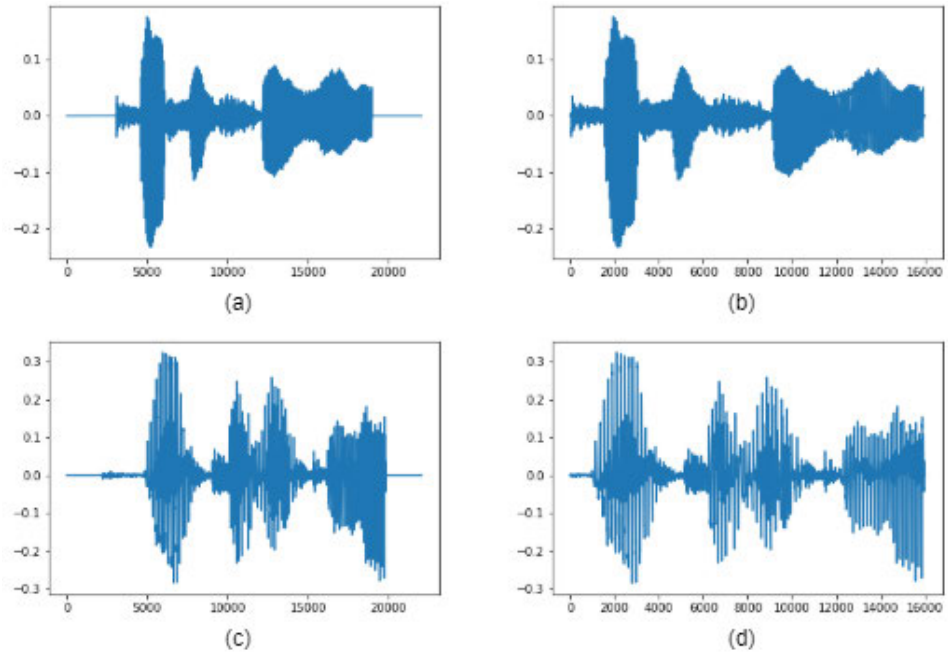



**FIGURE 4.** Speech signals (a) and (c) with their respective loudest sound segments in (b) and (d).

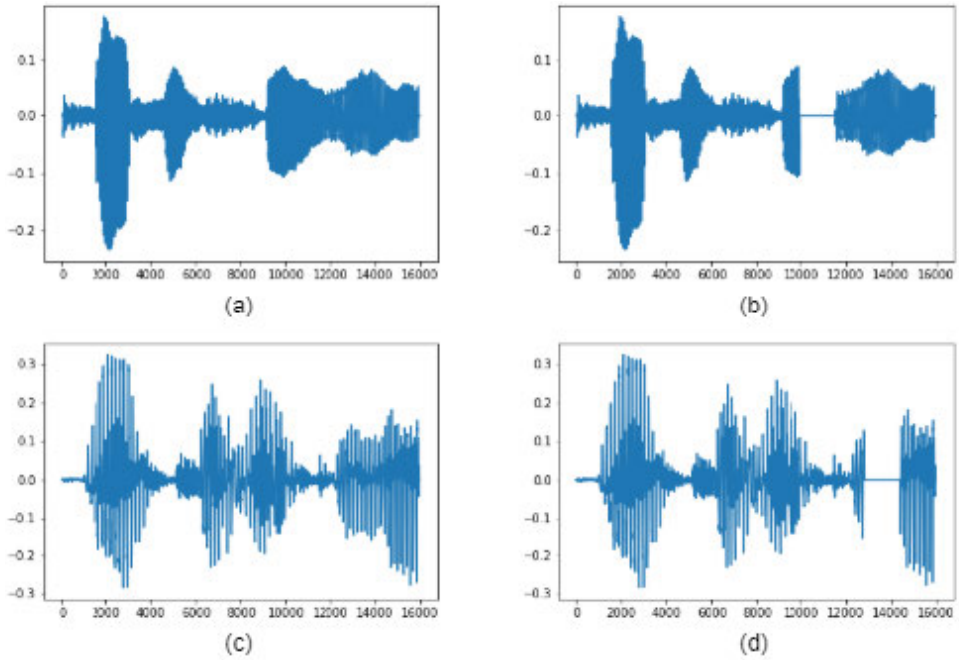



**FIGURE 5.** Speech signals (a) and (c) and respectively masked signals (b) and (d).

operations were applied compared with the application of any combination of operations.

### B. EXPERIMENTS ON SPEECH COMMAND DATASETS V1 AND V2

We first verified the performance of the proposed Wav2KWS model on Google Speech Command Datasets V1 [26] and V2 [27]. Dataset V1 comprises 10 commands, namely, yes, no, up, down, left, right, on, off, stop, and go as well as identifiers for silence and unknown commands. Dataset V2 adds 10 commands corresponding to the digits from zero to nine, thus totaling 22 commands.

For the experiments, we used the data split algorithm in [40] and divided the datasets in a ratio of 80:10:10 for

training:validation:test samples. Table 2 lists the characteristics of datasets V1 and V2.

**TABLE 2. Characteristics of Google Speech Command Datasets V1 and V2 used in this study.**

| Dataset version | No. samples | | | |
|---|---|---|---|---|
| | Training | Validation | Test | Total |
| V1 [26] | 22 236 | 3093 | 3081 | 28 410 |
| V2 [27] | 36 923 | 4445 | 4890 | 46 258 |

Table 3 lists the results of an experiment with the 12 shared commands in datasets V1 and V2. In dataset V1, the accuracy of the proposed Wav2KWS is only 0.1% higher than that of Res 8. Considering the number of model parameters in KWS, the Res 8 model provides a suitable performance. In addition, when compared with the MHAtt-RNN model, which shows the highest performance among the models evaluated on dataset V2, the accuracy of the proposed Wav2KWS model is 0.5% higher, and the proposed Wav2KWS model outperforms MatchboxNet 3 × 2 × 64 and TC-ResNet by 0.9% and 1.1% in accuracy, respectively.

**TABLE 3. Accuracy of baseline models and proposed Wav2KWS model on Google Speech Command Datasets V1 and V2 considering their 12 shared commands.**

| Model | Accuracy (%) | |
|---|---|---|
| | Dataset V1 | Dataset V2 |
| Att-RNN [14] | 95.5* | 96.9* |
| TC-ResNet [19] | 97.1* | 97.4* |
| MatchboxNet 3 × 2 × 64 [20] | 97.4* | 97.6* |
| Res 8 [34] | 97.8* | – |
| MHAtt-RNN [21] | 97.2* | 98.0* |
| Wav2KWS | **97.9** | **98.5** |

*Values obtained from the corresponding paper

Table 4 lists the experimental results on dataset V2 with its 22 commands. The proposed Wav2KWS model provides a 0.8% higher accuracy than the SOTA DenseNet-BiLSTM model. Although the Wav2KWS model is more computationally expensive, it can produce a robust high performance even for several commands from the Google Speech Command Datasets.

### C. EXPERIMENTS ON COMMANDS IN KOREAN RELATED TO SOLAR POWER SYSTEMS

To evaluate the performance of the proposed Wav2KWS model on KWS for commands in Korean, we selected 25 commands related to solar power systems and recorded 5738 samples per command from 12 Korean native speakers using their smartphones. To build a baseline model, 8250 samples were synthesized for 11 Korean speakers using the Naver CLOVA Speech Synthesis (CSS) platform (available at https://www.ncloud.com/product/aiService/css).

**TABLE 4. Accuracy of baseline models and proposed Wav2KWS model on Google Speech Command Dataset V2 with its 22 commands.**

| Model | Accuracy (%) |
|---|---|
| LSTM [42] | 93.7* |
| Att-RNN [14] | 94.5* |
| DenseNet-BiLSTM [18] | 96.6* |
| Wav2KWS | **97.8** |

* Values obtained from the corresponding paper

The characteristics of the recorded and synthesized speech samples are detailed in Table 5.

To verify the robustness of the proposed Wav2KWS model for unseen speakers, we divided the data differently. The data recorded by 10 out of the 12 speakers were used for training, and the data recorded by the other 2 speakers were equally split for validation and testing. Thus, 4738, 500, and 500 speech samples were used for training, validation, and testing, respectively.

In the solar speech command dataset, we prepared three different types of training data: only recorded data, only CSS synthesized data, and all data, obtaining the results listed in Table 6. As the human recorded samples are insufficient to train the model, the baseline model performance is relatively low compared with that obtained on the Google Speech Command Dataset.

The backbone in the proposed Wav2KWS model improves learning under the lack of diverse speech features in the existing speech recognition models, notably outperforming TC-ResNet and MHAtt-RNN by 11.1% and 10.9% in accuracy, respectively. Meanwhile, when only the synthesized CSS data are used, the accuracy of TC-ResNet and MHAtt-RNN substantially drops. In contrast, the proposed Wav2KWS model provides high accuracy regardless of the type of training data.

### D. EXPERIMENTS ON COMMANDS IN KOREAN RELATED TO HEALTHCARE AND DAILY ACTIVITIES

We used the automatically constructed dataset for experiments on a small Korean speech keyword dataset [43] collected from YouTube. The dataset consists of the most common keywords for healthcare and daily activities.

The keywords related to healthcare are vitamin (비타민), virus (바이러스), painkiller (진통제), diet (다어어트), side effect (부작용), protein (단백질), magnesium (마그네슘), aspirin (아스피린), energy (에너지), Tylenol (타이레놀), caffeine (카페인), stress (스트레스), hypertension (고혈압), and endoscope (내시경).

The keywords related to daily activities are Naver (네이버), calendar (캘린더), e-mail (이메일), camera (카메라), YouTube (유튜브), service (서비스), image (이미지), story (스토리), fine dust (미세먼지), navigation (네비게이션),

**TABLE 5. Number of speech samples recorded passively and synthesized by TTS corresponding to commands in Korean related to solar power systems.**

| Solar speech command dataset | | |
|---|---|---|
| Command | No. recorded samples | No. synthesized samples |
| Output control (출력 제어) | 229 | 330 |
| Frequency fluctuation (주파수 변동) | 229 | 330 |
| Failure time (장애 시간) | 230 | 330 |
| Angle control (각도 제어) | 229 | 330 |
| Active device (능동 디바이스) | 229 | 330 |
| Troubleshooting (고장 진단) | 229 | 330 |
| String inverter (스트링 인버터) | 229 | 330 |
| Generation efficiency (발전 효율) | 234 | 330 |
| Power generation comparison (발전량 비교) | 229 | 330 |
| Energy storage system (에너지 저장 장치) | 229 | 330 |
| Abnormal temperature (이상 온도) | 229 | 330 |
| Broadcast control (원격 관제) | 229 | 330 |
| Energy storage system (전력 저장 장치) | 229 | 330 |
| Drone (드론) | 230 | 330 |
| Module status (모듈 상태) | 229 | 330 |
| Monitoring (모니터링) | 234 | 330 |
| Integrated plate (집적판) | 229 | 330 |
| Comparison of power generation by zone (구역별 발전량 비교) | 229 | 330 |
| Solar module (태양광 모듈) | 229 | 330 |
| Visualization (시각화) | 229 | 330 |
| Central inverter (센트럴 인버터) | 230 | 330 |
| Output prediction (출력 예측) | 229 | 330 |
| Battery fire (배터리 화재) | 229 | 330 |
| Thermal imaging camera (열화상 카메라) | 229 | 330 |
| Generated output (발전 전력) | 229 | 330 |
| Total | 5738 | 8250 |

**TABLE 6. Accuracy of baseline models and proposed Wav2KWS model considering 25 commands from solar speech command dataset with TTS data.**

| Model | Accuracy (%) | | |
|---|---|---|---|
| | Only recorded data | Only CSS synthesized data | Recorded data and CSS synthesized data |
| TC-ResNet [19] | 88.5 | 45.4 | 93.2 |
| MHAtt-RNN [21] | 88.7 | 44.4 | 92.3 |
| Wav2KWS | 99.6 | **100** | **100** |

coffee shop (커피숍), mic (마이크), Kakao (카카오), message (메시지), and internet (인터넷).

After collecting the video URLs by using the YouTube Search API (Application Programming Interface; available at https://developers.google.com/youtube/v3), we downloaded the videos and their captions, and extracted the audio corresponding to the target keywords. To ensure the quality of the collected data, we selected the keywords correctly recognized by the Google Speech-to-Text API (available at https://cloud.google.com/speech-to-text/) to construct the dataset. As a result, the dataset contains 1065 samples, with 513 samples of 14 healthcare keywords and 552 samples of 15 daily activity keywords. As the dataset is small, we randomly divided it into training and test datasets at a ratio of 90:10 and excluded validation.

Table 7 indicates that the accuracy of the proposed Wav2KWS model reaches 100%, possibly because only data that the Google Speech-to-Text API can fully recognize were used. The MHAtt-RNN model also achieves 100% accuracy for keywords related to daily activities, but its accuracy is relatively low (approximately 90%) for keywords related to healthcare. Compared with MHAtt-RNN, TC-ResNet consistently achieves 98% accuracy on the two datasets. However, only the proposed Wav2KWS model provides a robust 100% accuracy on the evaluated dataset.

**TABLE 7. Accuracy of baseline models and proposed Wav2KWS model accuracy considering 14 keywords related to healthcare and daily activities.**

| Model | Accuracy (%) | |
|---|---|---|
| | Healthcare | Daily activities |
| TC-ResNet [19] | 98.1 | 98.1 |
| MHAtt-RNN [21] | 90.9 | **100** |
| Wav2KWS | **100** | **100** |

### *E. KWS ACCORDING TO NUMBER OF TRAINING SAMPLES*

Besides the abovementioned experimental results, we verified the effectiveness of the Wav2Vec 2.0 backbone in the proposed Wav2KWS model. When a dataset has more than 20 000 samples, the encoder is suitably trained, resulting in an accuracy over 90%, even if only one epoch is considered for transfer learning, as shown for different datasets in Fig. 6. The KWS accuracy quickly converges to a high value if more than 3000 training samples are used.

Fig. 7 shows the experimental results from applying the proposed Wav2KWS model under different numbers of speech samples per command in both Google Speech Command Dataset V2 with 12 keywords and the solar speech command dataset. With approximately 15 samples per command, Wav2KWS achieves 90% accuracy. With more than 40 recorded speech samples, Wav2KWS can easily achieve 99% accuracy. Moreover, the proposed Wav2KWS model converges to minimum boundary of loss in early step in the two datasets as in Fig. 8.

The experimental results suggest that KWS data required can be constructed using the Wav2Vec 2.0 backbone.

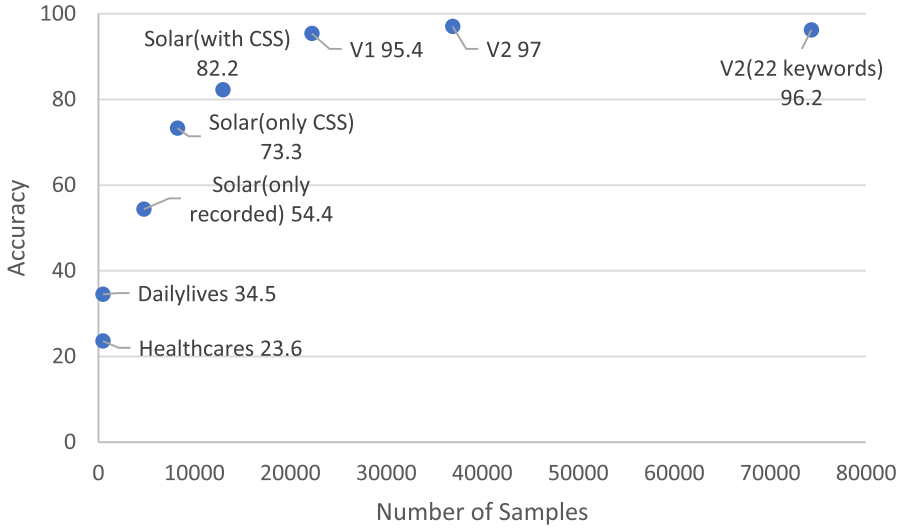
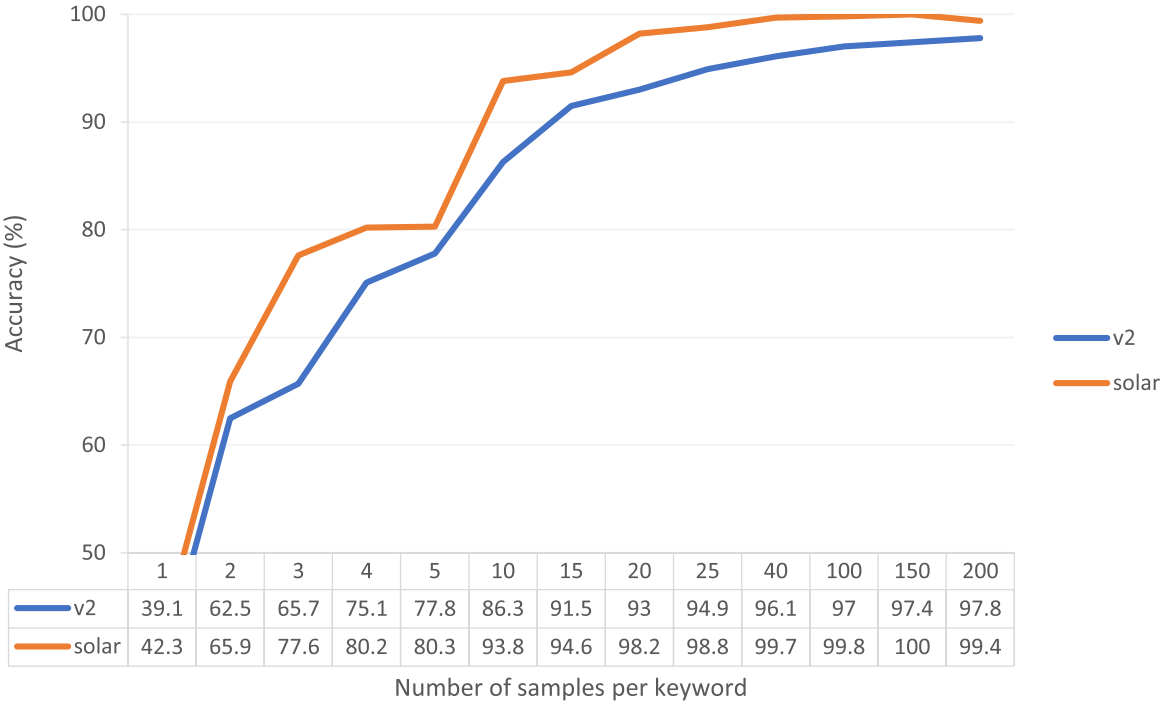



**FIGURE 6.** KWS accuracy of proposed Wav2KWS model when training for one epoch of transfer learning per dataset.




| | 1 | 2 | 3 | 4 | 5 | 10 | 15 | 20 | 25 | 40 | 100 | 150 | 200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| v2 | 39.1 | 62.5 | 65.7 | 75.1 | 77.8 | 86.3 | 91.5 | 93 | 94.9 | 96.1 | 97 | 97.4 | 97.8 |
| solar | 42.3 | 65.9 | 77.6 | 80.2 | 80.3 | 93.8 | 94.6 | 98.2 | 98.8 | 99.7 | 99.8 | 100 | 99.4 |

**FIGURE 7.** Accuracy according to number of utterances per class on Google Speech Command Dataset V2 and solar power system dataset.

Considering the high accuracy of the Wav2KWS model when training only with synthesized data (Table 6), data scarcity may be overcome to achieve robust KWS training in various languages.

To obtain 90% accuracy, the most recent head model in [8] requires 5–10 samples per command, but the proposed Wav2KWS model requires 10–15 samples. Nevertheless, to achieve more than 95% accuracy, Wav2KWS requires 40 samples for Google Speech Command Dataset V2 and 20 samples for the solar speech command dataset. In contrast, the head model requires approximately 125 training samples for these datasets.

If the minimum number of samples is available, the proposed Wav2KWS model achieves over 98% accuracy in datasets from various domains, and the pretrained Wav2Vec 2.0 backbone enables accurate KWS in other languages such as Korean.

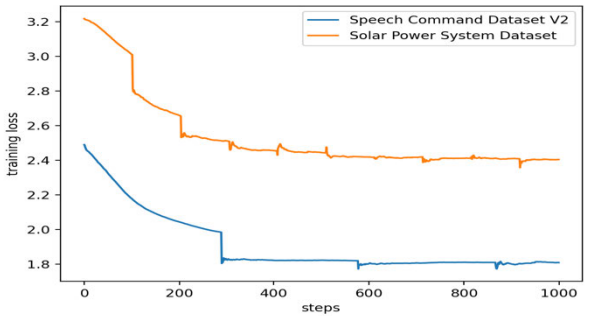



**FIGURE 8.** Learning curve on Google Speech Command Dataset V2 and solar power system dataset.

## V. CONCLUSION

We propose the Wav2KWS model that achieves robust and accurate KWS with few training samples. The model establishes a new SOTA performance on Google Speech Command Datasets V1 and V2 and three Korean speech command datasets. Existing studies have primarily used SOTA lightweight models or advanced deep learning architectures but require several hundreds of training samples. On the other hand, the proposed Wav2KWS model can obtain high KWS performance with a few available samples by leveraging a pretrained encoder, achieving perfect accuracy in specific cases. In detail, high-performance KWS with the proposed Wav2KWS model can be achieved with approximately 40 training samples in English and 20 training samples in Korean per command. Moreover, a pretrained model on an English corpus can be directly applied to Korean speech while maintaining a higher performance than baseline models. In future work, we plan to reduce the model weight by adopting distillation and model reduction techniques as in natural language processing. We also intend to implement the proposed Wav2KWS model on edge devices.

## REFERENCES

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4960–4964.

[2] D. Amodei, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 173–182.

[3] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "ContextNet: Improving convolutional neural networks for automatic speech recognition with global context," 2020, *arXiv:2005.03191*. [Online]. Available: http://arxiv.org/abs/2005.03191

[4] J. Pan, J. Shapiro, J. Wohlwend, K. J. Han, T. Lei, and T. Ma, "ASAPP-ASR: Multistream CNN and self-attentive SRU for SOTA speech recognition," 2020, *arXiv:2005.10469*. [Online]. Available: http://arxiv.org/abs/2005.10469

[5] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," 2019, *arXiv:1904.05862*. [Online]. Available: http://arxiv.org/abs/1904.05862

[6] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," 2020, *arXiv:2010.10504*. [Online]. Available: http://arxiv.org/abs/2010.10504

[7] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," 2020, *arXiv:2005.09629*. [Online]. Available: http://arxiv.org/abs/2005.09629

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[10] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. ICASSP*, 2014, pp. 1–5.

[11] S. Ö. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates, "Convolutional recurrent neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, Aug. 2017, pp. 1478–1482.

[12] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," 2017, *arXiv:1711.07128*. [Online]. Available: http://arxiv.org/abs/1711.07128

[13] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5484–5488.

[14] D. Coimbra de Andrade, S. Leo, M. Loesener Da Silva Viana, and C. Bernkopf, "A neural attention model for speech command recognition," 2018, *arXiv:1808.08929*. [Online]. Available: http://arxiv.org/abs/1808.08929

[15] S. Sigtia, R. Haynes, H. Richards, E. Marchi, J. Bridle, and S. Speech, "Efficient voice trigger detection for low resource hardware," in *Proc. Ininterspeech*, Oct. 2018, pp. 2092–2096.

[16] M. Sun, D. Snyder, Y. Gao, and V. K. Nagaraja, "Compressed time delay neural network for small-footprint keyword spotting," in *Proc. Interspeech* 2017, pp. 3607–3611.

[17] G. Tucker, "Model compression applied to small-footprint keyword spotting," in *Proc. Interspeech*, vol. 2016, pp. 1878–1882.

[18] M. Zeng and N. Xiao, "Effective combination of denseNet and BiLSTM for keyword spotting," *IEEE Access*, vol. 7, pp. 10767–10775, 2019, doi: 10.1109/ACCESS.2019.2891838.

[19] S. Choi, S. Seo, B. Shin, H. Byun, M. Kersner, B. Kim, D. Kim, and S. Ha, "Temporal convolution for real-time keyword spotting on mobile devices," 2019, *arXiv:1904.03814*. [Online]. Available: http://arxiv.org/abs/1904.03814

[20] S. Majumdar and B. Ginsburg, "MatchboxNet: 1D time-channel separable convolutional neural network architecture for speech commands recognition,"2020, *arXiv:2004.08531*. [Online]. Available: https://arxiv.org/abs/2004.08531

[21] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming keyword spotting on mobile devices," in *Proc. Interspeech*, 2020, pp. 3–7, doi: 10.21437/interspeech.2020-1003.

[22] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*. [Online]. Available: http://arxiv.org/abs/1804.03209

[23] J. Lin, K. Kilgour, D. Roblek, and M. Sharifi, "Training keyword spotters with limited and synthesized speech data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7474–7478, doi: 10.1109/ICASSP40776.2020.9053193.

[24] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "DetNet: A backbone network for object detection," 2018, *arXiv:1804.06215*. [Online]. Available: http://arxiv.org/abs/1804.06215

[25] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2014, pp. 3320–3328.

[26] *Speech Commands Dataset Version 1*. Accessed: Dec. 20, 2020. [Online]. Available: http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz

[27] *Speech Commands Dataset Version 2*. Accessed: Dec. 20, 2020. [Online]. Available: http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz

[28] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, Y.-H. Chen, S.-W. Li, and H.-Y. Lee, "Audio albert: A lite bert for self-supervised learning of audio representation," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 344–350.

[29] K. Clark, M.-T. Luong, G. Brain, Q. V. Le Google Brain, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," in *Proc. ICLR Conf.*, 2020, pp. 1–5. Accessed: Dec. 20, 2020. [Online]. Available: https://github.com/google-research/.

[30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2383–2392, doi: 10.18653/v1/D16-1264.

[31] P. Rajpurkar, R. Jia, and P. Liang, "Know what you Don't know: Unanswerable questions for SQuAD," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 784–789, doi: 10.18653/v1/p18-2124.

[32] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "SWAG: A large-scale adversarial dataset for grounded commonsense inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 93–104, doi: 10.18653/v1/d18-1009.

[33] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, Sep. 2019, pp. 3465–3469, doi: 10.21437/Interspeech.2019-1873.

[34] R. Tang, J. Lee, A. Razi, J. Cambre, I. Bicking, J. Kaye, and J. Lin, "Howl: A deployed, open-source wake word detection system," 2020, *arXiv:2008.09606*. [Online]. Available: http://arxiv.org/abs/2008.09606

[35] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1D time-channel separable convolutions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6124–6128, doi: 10.1109/ICASSP40776.2020.9053889.

[36] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*. [Online]. Available: http://arxiv.org/abs/1606.08415

[37] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011, doi: 10.1109/TPAMI.2010.57.

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

[39] *Extract Loudest Section*. [Online]. Available: https://github.com/petewarden/extract_loudest_section

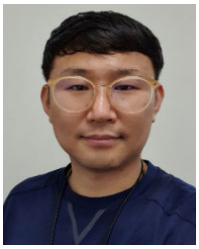
[40] *Split Algorithm*. [Online]. Available: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/speech_commands/input_data.py#L61

[41] D. S. Park, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Annu. Int. Speech Commun. Assoc. INTERSPEECH*, Sep. 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[42] R. Luo, "Multi-layer attention mechanism for speech keyword recognition," 2019, *arXiv:1907.04536*. [Online]. Available: http://arxiv.org/abs/1907.04536

[43] Y. Lim, D. Seo, J. Park, Y. Jung, C. Engineering, and K. National, "An automatic data construction approach for Korean speech command recognition," *J. Korea Soc. Comput. Inf.*, vol. 24, no. 12, pp. 17–24, 2019.

**DEOKJIN SEO** received the B.S. degree in computer engineering from the Kumoh National Institute of Technology (KIT), South Korea, in 2020, where he is currently pursuing the M.S. degree in computer engineering. His research interests include audio signal processing, natural language processing, and deep learning.



**HEUNG-SEON OH** received the B.S. degree in computer engineering from Korea Aerospace University (KAU), South Korea, in 2006, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2009 and 2014, respectively. He was a Senior Researcher with the Korea Institute of Science and Technology Information (KISTI), South Korea, from 2014 to 2018. Since 2018, he has been an Assistant Professor with the School of Computer Science and Engineering, Korea University of Technology and Education (KOREATECH), South Korea. His research interests include deep learning, computer vision, natural language processing, and information retrieval.

**YUCHUL JUNG** received the B.S. degree in computer science from Ajou University, South Korea, in 2001, and the M.S. degree in information and communication engineering and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 2005 and 2011, respectively. He has been working as an Assistant Professor with the Department of Computer Engineering, Kumoh National Institute of Technology (KIT), Gumi, since 2017. Before joining KIT, he worked as a Senior Researcher with the the Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, from 2009 to 2013, and the Korea Institute of Science and Technology Information (KISTI), from 2013 to 2017. His research interests include machine learning-based NLP (text mining, sentiment analysis, and automatic knowledge base construction), Korean speech recognition, and medicine 2.0. He has served as a Reviewer for *Knowledge-Based Systems*, *IEEE Communications Magazine*, and *ACM Transactions on Interactive Intelligent Systems (TiiS)*.