Low Resource Multi-ASR Speech Command Recognition

1st Isham Mohamed
Department of Computer Science and Engineering
University of Moratuwa
Katubedda 10400, Sri Lanka
jazeem.20@cse.mrt.ac.lk

2nd Uthayasanker Thayasivam
Department of Computer Science and Engineering
University of Moratuwa
Katubedda 10400, Sri Lanka
rtuthaya@cse.mrt.ac.lk

2022 Moratuwa Engineering Research Conference (MERCon) | 978-1-6654-8786-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/MERCon55799.2022.9906230

***Abstract*—There are several applications when comes to spoken language understanding (SLU) such as topic identification and intent detection. One of the primary underlying components used in SLU studies are ASR (Automatic Speech Recognition). In recent years we have seen a major improvement in the ASR system to recognize spoken utterances. But it is still a challenging task for low resource languages as it requires 100's hours of audio input to train an ASR model. To overcome this issue recent studies have used transfer learning techniques. However, the errors produced by the ASR models significantly affect the downstream natural language understanding (NLU) models used for intent or topic identification. In this work, we have proposed a multi-ASR setup to overcome this issue. We have shown that combining outputs from multiple ASR models can significantly increase the accuracy of low-resource speech-command transfer-learning tasks than using the output from a single ASR model. We have come up with CNN based setups that can utilize outputs from pre-trained ASR models such as DeepSpeech2 and Wav2Vec 2.0. The experiment result shows an 8% increase in accuracy over the current state-of-the-art low resource speech-command phoneme-based speech intent classification methodology.**

***Index Terms*—Speech Intent Classification, Low-Resource, DeepSpeech2, Wav2Vec 2.0, Tamil**

I. Introduction

With the advancement of voice-enabled technologies, spoken language understanding has evolved so much now it is almost as good as a real human. There are various applications when it comes to Spoken Language Understanding(SLU) including Speech command recognition and topic identification [1]–[3]. The common approach to SLU is to first use an automatic speech recognition(ASR) model to transcribe the voice and use a natural language understanding(NLU) model train on the text corpus to do various classification tasks.

Recently end-to-end ASR models like DeepSpeech2 [4] and Wav2Vec 2.0 [5] have been introduced as an alternative to previously predominant solutions based on Hidden Markov Models such as Kaldi [6]. However, these deep neural network-based models required an enormous amount of audio data during the training. For example, the original DeepSpeech2 implementations were trained on more than 7000 hours of data and the new DeepSpeech2 was trained on 11,000+ hours of data. Only a few mainstream languages have these kinds of large data-set, to begin with, such as English and Mandarin. But these models won't perform that well even with popular languages such as Swiss and German which have around 100 hours of data each [7].

But when it comes to intent detection and topic modeling tasks, we don't necessarily need to have the complete transcript, rather we only need a way to model the utterance in a way so that this can be used by the downstream NLU models. There many transfer learning studies have been conducted [9], [10], [12]. The main fundamental approach used in these studies is to use the scriptions or N-hypothesis produced by the ASR model as the feature of an NLU model. However, the performance of these NLU models greatly depends on how accurate the transcription is provided by the ASR model.

In this study, we have proposed a way to reduce the effect of the ASR model on the downstream model by introducing ways to combine multiple ASR features to a single CNN NLU model. During our experiment, we have observed significant improvement over using a single ASR model and reported state-of-the-art accuracy of 88.25% for Tamil speech to command recognition.

II. Related Works

When it comes to speech-command recognition, the typical approach is to use an acoustic model which is directly trained with the annotated domain data. [13] authors have used CTC-based LSTM acoustic models to train on approximately 2,000 hours of mobile-based google voice search traffic. It is not possible to find this volume of data when it comes to all the domains.

In such scenarios where there are not much domain-specific speech data, transcripts generated by ASR models are widely used to produce the text feature of the voice and these features are used in downstream models to identify the intent or topi [3], [15]. Building an ASR model is not viable for most of the time, the sheer volume of data required to train those models is so high, only a few mainstream language models are so far built. This is one of the main drawbacks when it comes to low-resource languages.

To address the low-resource ASR problems, transfer learning ASR [9], [20] and multilingual transfer learning ASR [14] are explored via using different source languages to improve the performance of low-resource languages. Recent works have focused on coming up with such low resource

978-1-6654-8786-3/22/$31.00 ©2022 IEEE

systems by transfer learning techniques such as leveraging the intermediate outputs of English based DeepSpeech2 ASR model as the input feature for the downstream NLU models [9] and further improved by using phoneme based ASR model trained by the English language and CNN based model as the NLU model [16]. In this research, authors have tried to use these techniques to improve the SLU for Tamil and Sinhala language archiving 81% and 97% accuracy respectively. A phoneme is any of the perceptually distinct units of sound in a specified language that distinguishes one word from another.

ASR models can be divided into two categories based on the output features, character-based ASR models and phoneme-based ASR models. A character-based ASR model [4], [5] is trained to produce a probability distribution P_t(c) over vocabulary characters c per each time step t. Where in a phoneme-based ASR model [21] it is trained to produce a probability distribution of P_t(p) over a vocabulary sounds per each time step t. As phonemes are the smallest units of sound in a language, the expectation is for the ASR transcription to be more similar to the correct utterance at the phoneme level than at the character or word levels. This is indeed proved by the study [16] by improving the accuracy over [9]

DeepSpeech2 [4] converts the input speech into Melspectrograms, then applies CNN and RNN, and finally outputs the text using Connectionist Temporal Classification (CTC). Connectionist Temporal Classification (CTC) is a method often used in character recognition and speech recognition, in combination with LSTM and RNN. In character and speech recognition, the width of a single character and the time length of a single phoneme is variable. This method solves the problem of variable width and time length of a phoneme by erasing the same character in succession on the decoder side.

Wav2Vec 2.0 [5] is one of the current state-of-the-art models for ASR. The model leverages self-supervised training which is quite a new concept in this field. It is trained in two phases. The first phase is in a self-supervised mode, which is done using unlabeled data and aims to achieve the best speech representation possible. The second phase of training is supervised fine-tuning, during which labeled data is used to teach the model to predict particular words or phonemes.

When it comes to combining features from multiple sources, it is often common to use dual input models [17], [18]. In these studies, the authors have used 2 different CNN models to learn the features from 2 different inputs and then combine the features into a single CNN model which is used for classification purposes. These types of CNN models are used in scenarios where there need to be multiple sources of inputs need to be combined. For example, combining voltage trend and current flow trend, or combining pictures taken from multiple angles.

In one similar study [19] the authors have improved Speech Emotion Recognition (SER) by combining the magnitude spectrogram with the modified group delay spectrogram, by including synthetic noise in the training data. Although we have not used spectrogram in this study to improve the SLU, the idea of combining multiple inputs from a different source as a means to improve the overall performance was an inspiration for this study.

The main approaches previous studies have taken to address low-resource speech intent identifications are to build a low-resource ASR model or retrain some of the layers of the pre-trained ASR models via transfer learning or use a pre-trained asr model to generate N-hypothesis and use an NLU model to classify top of it. Although studies have primarily used a single ASR model at a time, we have seen in similar other domains such as SRE, authors have tried to combine features from multiple upstream to train a single downstream SRE model. This is not yet been tried when it comes to combining the features from multiple ASR models to train a single NLU model. In the next section, we propose 2 main approaches to do so.

III. Methodology

All the previous researches mainly focus on using a single ASR model to predict either a word sequence or phonemic sequence which is then to be used in an NLU model. This would mean that any error produced by the ASR model will be propagated down to the NLU model thus affecting its performance of it. In this study, we focused more on improving the performance of NLU models not just by using a better ASR model to transfer learning from, but to reducing the error propagation by combining different ASR models.

We propose 2 different ways where we can combine the character probability sequence provided by 2 different character-based ASR models, DeepSpeech2 [4] and Wav2Vec 2.0 [5]

A. Multi-ASR Combinations

using the above two ASR models, we generated features independently. one of the main differences we observe apart from the difference in the character space of those 2 algorithms, is the length of the mapping. where DeepSpeech2 outputs a longer character probability sequence of max of 555, Wav2Vec 2.0 only produced up to 256 character sequences. We have proposed 2 ways of combining features learned from multiple ASR models.

1) Method 1 - Dual-input CNN Models: There are previous researches [9], [16], [20] done to identify a fixed set of intents using the features extracted from audio inputs. These mainly used a single ASR model to extract the feature from the audio input and used classification models such as Support Vector Machine (SVM) , Feed-forward Neural Networks(FFN), and Convolution Neural Networks (CNN). Out of all the studies [16] has shown a state-of-the-art accuracy using 1D and 2D CNN models. In 1D type, the convolution and pooling operations are done along only on one dimension, while in 2D this happens along on two directions.

In this study, we mainly focus on combining the ASR features from DeepSpeech2 and Wav2Vec 2.0 ASR models. The first method we tried is to train 2 2D CNN models with each feature and train the last layers by combining the models

and applying a softmax layer to identify the classification. "Fig. 1" explains the overall architecture of the model.

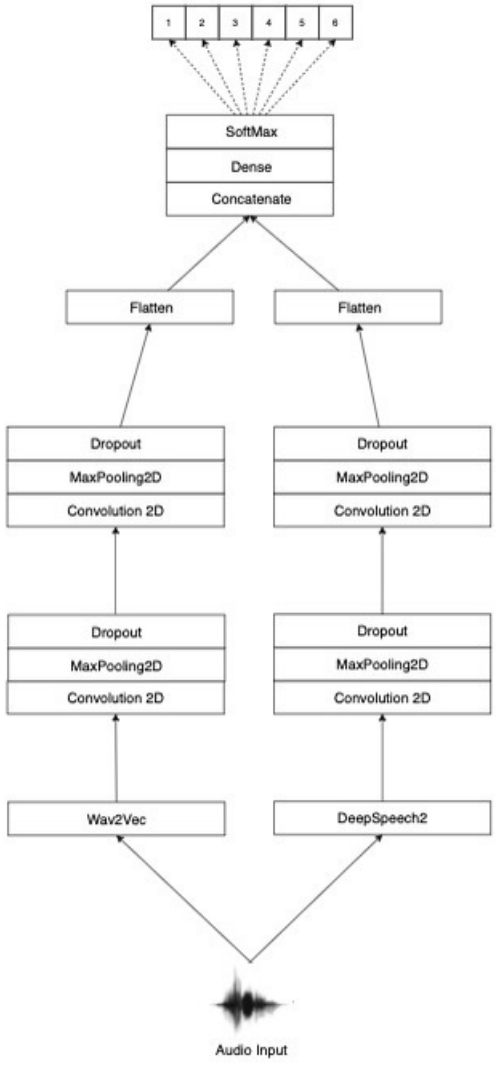

Fig. 1. Overview of dual-input CNN model architecture

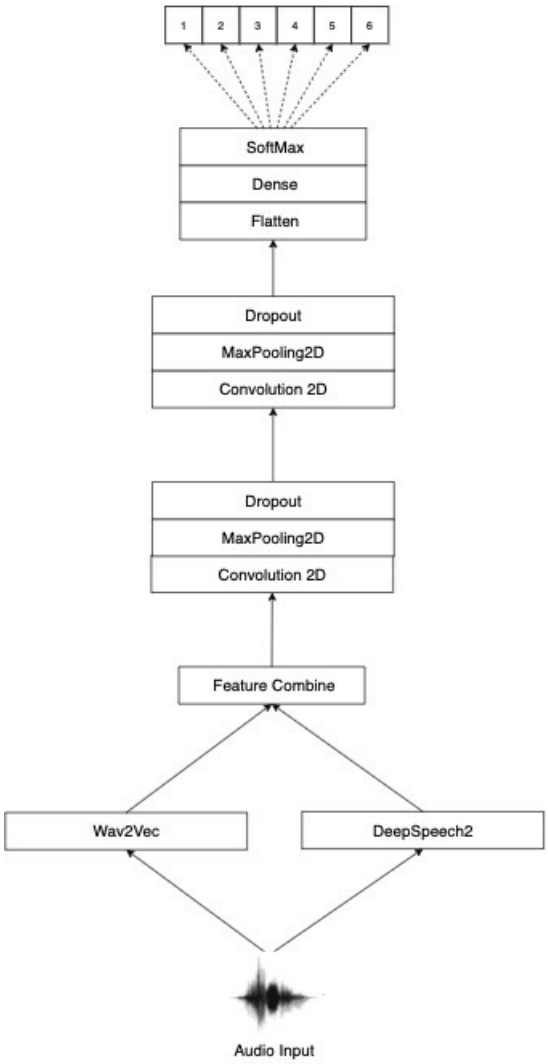

Fig. 2. Overview of combined-input model architecture

2) Method 2 - Feature Combine : First, we tried the obvious, which is a concatenation of the 2 features into a single feature. When it comes to CNN it is important to have the features uniform in dimensions, to tackle this issue we initially tried adding padding of zeros to make every probability sequence has the same dimension. Then we used a 2D-CNN neural network explained in "Fig. 2" to treat the concatenated feature as a single feature do the classification.

The feature combine module of the combined-input model is a simple concatenation of NumPy arrays. For DeepSpeech2, if a transcribed probability sequence of an audio clip is $DS2_{act\_length}$ in length, we add $DS2_{max\_length} - DS2_{act\_length}$ zeros to each character probability sequence such that the length of a given character sequence has the same dimension for each audio clip, $(DS2_{chr\_size}, DS2_{max\_length})$. Similarly, for Wav2Vec 2.0, if a transcribed probability sequence of an audio clip is $W2V_{act\_length}$ in length, we add $W2V_{max\_length} - W2V_{act\_length}$ zeros to each character probability sequence such that the length of a given character sequence has the same dimension for each audio clip, $(W2V_{chr\_size}, W2V_{max\_length})$. The Wav2Vec 2.0 model we used have 32 characters and the DeepSpeech2 model we used has 29 character which creates a difference in number of rows in each feature output from Wav2Vec 2.0 and DeepSpeech2. To overcome the difference in the row count for each sets of features, we again used zero padding to the DeepSpeech2 features such as a feature would have a dimension of $(W2V_{chr\_size}, DS2_{max\_length})$ so that it matches the character size of Wav2Vec 2.0 models. Once the DeepSpeech2 character size is adjusted, we combined both feature so that a combined feature will be $(W2V_{chr\_size}, DS2_{max\_length} + W2V_{max\_length})$ in dimension. "Fig. 3" illustrates the dimension of the final combined feature of a given audio clip.

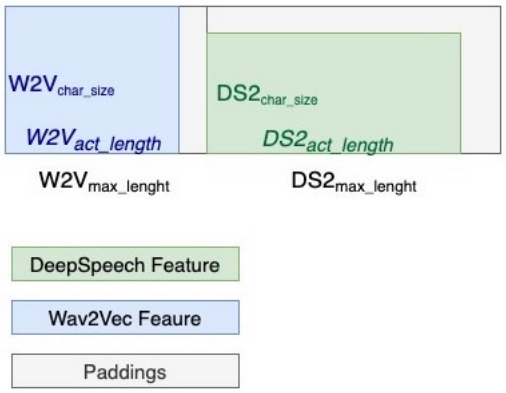

Fig. 3. DeepSpeech2 Wav2Vec 2.0 Feature Combine

IV. Data set

The data set is used in previous experiments [9], and improved in [16]. "TABLE I" briefly explains the data set. This data set contains only 400 entries added up to only 0.5 hours of training data. The primary language of this dataset is Tamil, although it is code-mixed with some English terms. The audio clips are collected via mobile phone microphones, which allows us to simulate real-world scenarios.

TABLE I
Details of the dataset

Intent	Inflections	Samples
1. Request Acc. Balance	7	101
2. Money Deposit	7	75
3. Money Withdraw	5	62
4. Bill Payments	4	46
5. Money Transfer	4	49
6. Credit Card Payments	4	67
Total	31	400
Unique Words	46	
Size in Hours	0.5	
Number of Speakers	40	

V. Experiment

For this experiment, we used pre-trained DeepSpeech2 model [4] with 11% WER on LibriSpeech corpus and Wav2Vec 2.0 model [5] with 7.4% WER on the same LibriSpeech corpus. We also used the phoneme-based model used in [16] for bench-marking our works.

These ASR models are used to extract the character probability distribution of each audio clip. Then this character probability distribution is used as the input for the CNN models which will classify the intent of the original audio clip.

We first experiment with only using one single ASR model each time with the CNN model benchmark the performance of the NLU model when each of these ASR models is used alone. Afterwards, we combined features extracted DeepSpeech2 and Wav2Vec 2.0 models with the proposed architecture in the methodology section 3. The "TABLE II" shows the details of the experiments.

TABLE II
Details of the experiments

Experiment No.	ASR Model(s)	NLU Model
Exp. 1	Phoneme	1D CNN
Exp. 2	DeepSpeech2	2D CNN
Exp. 3	Wav2Vec 2.0	1D CNN
Exp. 4	Wav2Vec 2.0	2D CNN
Exp. 5	DeepSpeech2 + Wav2Vec 2.0 (Combined Feature)	2D CNN
Exp. 6	DeepSpeech2 + Wav2Vec 2.0	Dual-Input CNN

To find the accuracy we used 5-fold cross-validation since the data entry we have is very low. We optimized only a selected number of parameters of the CNN models such as the number of filters, kernel size, and the dropout rate for each layer. We used the Bayesian search algorithm for our hyper-parameter tuning.

VI. results

TABLE III
Details of the experiment results

Experiment No.	ASR Model(s)	NLU Model	Accuracy
Exp. 1	Phoneme	1D CNN	81.35%
Exp. 2	DeepSpeech2	2D CNN	76.30%
Exp. 3	Wav2Vec 2.0	1D CNN	43.20%
Exp. 4	Wav2Vec 2.0	2D CNN	71.62%
Exp. 5	DeepSpeech2 + Wav2Vec 2.0 (Combined Feature)	2D CNN	**88.25%**
Exp. 6	DeepSpeech2 + Wav2Vec 2.0	Dual-Input CNN	83.50%

The "TABLE III" shows the results obtained from the above sets of experiments. The best performance we observed is from the 2D-CNN model fed with combined features from DeepSpeec2 and Wav2Vec 2.0. We were able to archive 88.25% accuracy with this model which is higher than the current state-of-the-art performance 81.35%. The proposed Dual-Model setup was also able to provide a higher accuracy than the current state-of-the-art solution even though it is not a significant improvement. If either DeepSpeech2 or Wav2Vec 2.0 model is used to transfer learn from, the NLU model is performing poorly than the phoneme based solution. This leads us to believe the improvement we observed in the setup is not just because of a better ASR model is been used to transfer learn from, but the technique we used combine the features generated from multiple ASR models.

TABLE IV
DIFFERENCE IN THE MODEL TRANSCRIPTIONS FOR A GIVEN UTTERANCE
(DS2 - DEEPSPEECH2, W2V - WAV2VEC 2.0)

Utterance	DS2 Transcription	W2V Transcription
Kācu innoru account ku mātta vēṇum (Money Transfer to another account)	causi nodid gon o coman theworom	cause he not i goin to gome out o un em
	care nnowi con ocomat_thean	caze in nore egon do cuma ta vonu
	casly an moda cound the mot thhe ra	cassly innude countemate wen
	cary nnodicont ccommatevernnam	carsi noricont comatavurno
	casi nnoo goncamatowerrn	casi norecon do gomant da verno

VII. DISCUSSION

In this set of experiments, we were able to replicate the work and the accuracy of the current state-of-the-art method found in the previous study [16]. Apart from that, we were also able to produce similar performance from DeepSpeech2 character-based ASR models mentioned in the same study. When we tried to evaluate the performance of Wav2Vec 2.0, which is similar to the DeepSpeech2, we got almost similar results as DeepSpeech2. This validates that the improvement in the accuracy we got in experiments 5 and 6 is not just because of a better ASR model wa used.

When we look into experiments 5 and 6, we can see an improvement over experiments 2 - 3 which only used a single ASR model, whereas experiments 5 and 6 used 2 ASR models. This indicates that the CNN model could learn additional features by combining the output from DeepSpeech2 and Wav2Vec 2.0 models.

The "TABLE IV" explains the encoded character sequence produced by DeepSpeech2 and Wav2Vec 2.0 ASR models for a code-mix utterance. As we can see the predicted character sequence is completely different from each other. One common difference we observed is that DeepSpeech2 was able to capture similar-sounding English terms at the start of the sentence, but towards the end of the sentence, the predicted terms do not sound similar to the original audio at all. Meanwhile, the Wav2Vec 2.0 was able to predict the terms sounding similar to the audio towards the end of the sentence but at the start of the sentence, it does not output similar sounding as accurate as DeepSpeech2. This is one of the main reasons why these features output from the 2 models are complementary to each other and the model was able to learn more rich features when both features were provided.

One other important outcome of we observed was that experiment 5 had higher accuracy than experiment 6. Experiment 5 was "Method 2 - Feature Combined" and experiment 6 was "Method 1 - Dual-input CNN Models" explained in section 2. We observed that combining features before feeding to the CNN model increased the accuracy of the CNN model significantly than using a Dual-input CNN model. This can be due to the lower number of training, Since the dual-input CNN model has separate branching for individual features, it introduces more layers than the original CNN model in experiment 5. We need more data, at least more than 1000 entries to train these layers based on the study [16].

Overall, models which learned from the combined feature produced 2 ASR models, outperformed the state-of-the-art phoneme-based single ASR setup. We observed lower accuracy when any of the ASR models were used in a single ASR setup.

VIII. CONCLUSION AND FUTURE WORKS

In this paper, we presented a state-of-the-art setup for low-resource SLU tasks such as speech command recognition and topic identification. We were able to report 88.25% accuracy with less than 0.5 hours of Tamil audio clips. We identified that combining the ASR model in the situation where the learned features are complementary, could significantly increase the accuracy of NLU models in low resource-transfer learning setups.

In the future, we are hoping to apply this setup to different domains as well as different low-resource languages. We also expanding this multi ASR model to combine more than 2 ASR models as well as phoneme-based ASR models.

REFERENCES

[1] Liu, Chunxi, Jan Trmal, Matthew Wiesner, Craig Harman, and Sanjeev Khudanpur. "Topic identification for speech without asr." arXiv preprint arXiv:1703.07476 (2017).

[2] Ram, Ashwin, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn et al. "Conversational ai: The science behind the alexa prize." arXiv preprint arXiv:1801.03604 (2018).

[3] Chen, Yuan-Ping, Ryan Price, and Srinivas Bangalore. "Spoken language understanding without speech recognition." In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6189-6193. IEEE, 2018. Processing (ICASSP). IEEE, 2018, pp. 6189–6193.

[4] Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." In International conference on machine learning, pp. 173-182. PMLR, 2016.

[5] Yi, Cheng, Jianzhong Wang, Ning Cheng, Shiyu Zhou, and Bo Xu. "Applying wav2vec2. 0 to speech recognition in various low-resource languages." arXiv preprint arXiv:2012.12121 (2020).

[6] Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann et al. "The Kaldi speech recognition toolkit." In IEEE 2011 workshop on automatic speech recognition and understanding, no. CONF. IEEE Signal Processing Society, 2011.

[7] Agarwal, Aashish, and Torsten Zesch. "Ltl-ude at low-resource speech-to-text shared task: Investigating mozilla deepspeech in a low-resource setting." In SwissText/KONVENS. 2020.

[8] Young, Matt. Technical writer's handbook. University Science Books, 2002.

[9] Karunanayake, Yohan, Uthayasanker Thayasivam, and Surangika Ranathunga. "Transfer learning based free-form speech command classification for low-resource languages." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pp. 288-294. 2019.
[10] Kunze, Julius, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. "Transfer learning for speech recognition on a budget." arXiv preprint arXiv:1706.00290 (2017).
[11] Misbullah, Alim, Kurnia Saputra, and Fauzy Nisa. "Customized Acoustic Model using Low-Resource Indonesian Speech Dataset for Short Command Speech Recognition System." In 2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), pp. 176-180. IEEE, 2021.
[12] Dinushika, Thilini, Lakshika Kavmini, Pamoda Abeyawardhana, Uthayasanker Thayasivam, and Sanath Jayasena. "Speech command classification system for sinhala language based on automatic speech recognition." In 2019 International Conference on Asian Language Processing (IALP), pp. 205-210. IEEE, 2019.
[13] McGraw, Ian, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif et al. "Personalized speech recognition on mobile devices." In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5955-5959. IEEE, 2016.
[14] Wang, Changhan, Juan Pino, and Jiatao Gu. "Improving cross-lingual transfer learning for end-to-end speech recognition with speech translation." arXiv preprint arXiv:2006.05474 (2020).
[15] Lugosch, Loren, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. "Speech model pre-training for end-to-end spoken language understanding." arXiv preprint arXiv:1904.03670 (2019).
[16] Karunanayake, Yohan, Uthayasanker Thayasivam, and Surangika Ranathunga. "Sinhala and tamil speech intent identification from english phoneme based asr." In 2019 International Conference on Asian Language Processing (IALP), pp. 234-239. IEEE, 2019.
[17] Chong, Thern Chang, Nien Loong Loo, Yeong Shiong Chiew, Mohd Basri Mat-Nor, and Azrina Md Ralib. "Classification Patient-Ventilator Asynchrony with Dual-Input Convolutional Neural Network." IFAC-PapersOnLine 54, no. 15 (2021): 322-327.
[18] Sun, Sukkyu, Ahnul Ha, Young Kook Kim, Byeong Wook Yoo, Hee Chan Kim, and Ki Ho Park. "Dual-input convolutional neural network for glaucoma diagnosis using spectral-domain optical coherence tomography." British Journal of Ophthalmology 105, no. 11 (2021): 1555-1560.
[19] Wijayasingha, Lahiru, and John A. Stankovic. "Robustness to noise for speech emotion classification using CNNs and attention mechanisms." Smart Health 19 (2021): 100165.
[20] Kunze, Julius, Louis Kirsch, Ilia Kurenkov, Andreas Krug, Jens Johannsmeier, and Sebastian Stober. "Transfer learning for speech recognition on a budget." arXiv preprint arXiv:1706.00290 (2017).
[21] Lugosch, Loren, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. "Speech model pre-training for end-to-end spoken language understanding." arXiv preprint arXiv:1904.03670 (2019).