Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.022021
Article

Tech Science Press

End-to-End Speech Recognition of Tamil Language

Mohamed Hashim Changrampadi[1,*], A. Shahina[2], M. Badri Narayanan[2] and A. Nayeemulla Khan[3]

[1]Department of Electronics and Communication Engineering, C.Abdul Hakeem College of Engineering & Technology, Melvisharam, 632509, India
[2]Department of Information Technology, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, 603110, India
[3]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, India
*Corresponding Author: Mohamed Hashim Changrampadi. Email: hashim@alumni.chalmers.se

Received: 25 July 2021; Accepted: 22 September 2021

Abstract: Research in speech recognition is progressing with numerous state-of-the-art results in recent times. However, relatively fewer research is being carried out in Automatic Speech Recognition (ASR) for languages with low resources. We present a method to develop speech recognition model with minimal resources using Mozilla DeepSpeech architecture. We have utilized freely available online computational resources for training, enabling similar approaches to be carried out for research in a low-resourced languages in a financially constrained environments. We also present novel ways to build an efficient language model from publicly available web resources to improve accuracy in ASR. The proposed ASR model gives the best result of 24.7% Word Error Rate (WER), compared to 55% WER by Google speech-to-text. We have also demonstrated a semi-supervised development of speech corpus using our trained ASR model, indicating a cost effective approach of building large vocabulary corpus for low resource language. The trained Tamil ASR model and the training sets are released in public domain and are available on GitHub.

Keywords: End to end speech recognition; deep learning; under-resourced language; semi-supervised speech corpus development

1 Introduction

The recent advancement in Automatic Speech Recognition (ASR) in the past couple of years is commendable, surpassing even human perception. However, most of these achievements are limited to languages with massive digital resources. Low resource languages always have challenges in adopting similar methodology due to limitations or unavailability of enormous training data, pronunciation dictionaries, language model, etc. In addition, low resource languages have their own challenges like code-switching, less fluent native speakers for data collection, and too many dialects. In this paper, we investigate the use of open-source speech recognition toolkits to build a speech recognition model for the Tamil language. This developed pre-trained model will provide an out-of-the-box support for transfer learning for keyword spotting, isolated word recognition, etc.

In the English language, speech recognition research has witnessed massive speech corpora and state-of-the-art ASR systems. Some of the common speech corpora in the English language include Switchboard (300 h), LibriSpeech (960 h), TedLium-3 (450 h), Common Voice (1400 h) and SPGISpeech (5000 h) [1]. Apart from Switchboard, all other stated English corpora are available for free download and use for research and non-commercial purposes. The speech recognition accuracy in terms of Word Error Rate (WER) for English has improved significantly in recent years. The WER on LibriSpeech test-clean [2] dataset has improved from 5.33 [3] to 1.4 [4]. However, low-resourced Indian languages lack similar massive speech corpus for ASR research. Recently, Microsoft [5] released a dataset for speech recognition challenge for low resource Indian languages. The dataset consists of 50 h of transcribed speech in three Indian languages-Tamil, Telugu, and Gujarati, amounting to a total of 150 h of data, with 40 h of training data and 5 h of test data for each language. Several attempts were also made to develop massive speech corpus for under-resourced languages using augmentation [6]. Tab. 1, summarizes the corpus and recent works in Tamil ASR, stating that almost all the corpus used are not available in public domain. Therefore, there is a tremendous need to develop a massive Tamil language speech corpus or a pretrained model to assist in transfer learning, semi-supervised corpus development, etc.

In this paper, we present a novel approach to build a pre-trained model using low resources and substantially assist in developing a massive speech corpus using semi-supervised learning. To our knowledge, this is the first attempt to use the Common Voice dataset and release a pre-trained ASR model for Tamil language.

The major contributions of this paper are documented below:

- **Open-source ASR Model:** Although pre-trained ASR models are available for most languages, they are seldom available for under-resourced languages. Use of a pre-trained model will ease transfer learning approach and lessen the complexity of training the speech model from scratch. In this paper, we have developed Tamil ASR and released the trained model for transfer learning. We have demonstrated isolated digit recognition using transfer learning from our trained Tamil ASR model.
- **Semi-Supervised Speech Corpus Development:** Large vocabulary corpus is essential for development of generic speech recognition systems. The process is time consuming and costly. However, in our paper, we have validated the invalidated set of Common Voice Tamil dataset using our trained ASR model. The validation of speech dataset using semi-supervised assistance has reduced time by 75% compared to manual validation. Such a semi-supervised approach could escalate the process of building a large vocabulary corpus.
- **Cost-effective Approach:** Solving any machine learning problem requires massive amount of data and extensive training of data. Training any speech recognition model depends on computational and financial resources. Since these resources are not available in a financially constrained environment, an alternative approach to achieve state-of-the-art results is necessary. In this paper, we have fully trained our ASR model using open-source toolkits, open dataset, and free computing resources like Google Colab. We have demonstrated that using available resources on the internet, design and development of ASR for under-resourced languages could be performed in a cost-effective manner.

The outline of the paper is as follows: Section 2 reviews related works done in Tamil ASR and challenges of Tamil language. Section 3 describes the processes in the proposed ASR system; development of speech corpus, ASR architecture and language modelling. Section 4 explains the training setup with experimentation results and demonstrated isolated digit recognition using transfer learning. Finally, Section 5 presents our conclusion.

2 Tamil ASR: Related Works

Automatic Speech Recognition (ASR) has progressed extensively in recent years with state-of-the-art results in the English language. End-to-End speech recognition systems [3] have eliminated the need for preprocessing of audio data, without compromising recognition accuracy. The sequence-to-sequence model with attention approach [7] has also produced promising results. Similar attention-based approach in Speech Emotion Recognition (SER) [8] has shown impressive performance. Convolutional Long Short Term Memory (ConvLSTM) Network [9] originally proposed for forecasting rainfall, works well for speech recognition and SER systems [10]. Convolutional Neural Network (CNN) based deep learning models have achieved promising results in ASR and SER systems [11,12]. However, most of the ASR systems developed for the English language have used massive corpus for training, which makes it difficult for under-resourced languages to replicate such techniques for developing ASR systems.

Speech recognition systems for Tamil language are seldom built using single clean large corpus or as a baseline monolingual end-to-end system. Often, transfer learning [13,14], data augmentation [6], language adaptation [15] or limited vocabulary [13,16,17] is opted. Transfer learning based Tamil ASR [14] produced better WER of 46.2%, than the baseline ASR with WER of 49.9%. Due to unavailability of large Tamil corpus, a monolingual Tamil ASR developed by massive augmentation of Tamil speech corpus using speed and volume perturbation, SpecAugment and addition of noise, gave an average of 47.6 WER [6]. Language adaptation is the process of pre-training in a source language and then training in a target language. Pretrained models of Bengali, Tagalog and Zulu were used to perform language adaptation for Tamil monolingual ASR that resulted in Character Error Rate (CER) of 48.0 [15].

Majority of these works are not close enough for real-world applications because of high WER/CER, and hence there is need for exhaustive research to develop a state-of-the art Tamil ASR. One major step towards this is to develop a massive Tamil speech corpus. Tamil language is considered as one of the oldest languages, yet focus on ASR research is minimal due to many challenges and complexity of the language.

Table 1: Overview of Tamil ASR systems

S. No.	Duration (#hours)	Vocab size	Publicly available	Technique	WER	Ref.
1	0.5	-	No	1D and 2D CNN	20%	[13]
2	150	75 k	No	Time Delay Neural Network (TDNN)-Hidden Markov Model (HMM)	17%	[18]
3	-	3 k	No	Bidirectional Recurrent Neural Network (B-RNN)	-	[16]
4	6.5	13 k	No	Deep Neural Network (DNN)-HMM	3.50%	[17]
5	45	-	No	LSTM	19.59%	[5]
6	50	-	No	TDNN, Bidirectional LSTM (BLSTM)	13.92%	[19]
7	-	-	No	CNN	48.0% (CER)	[15]
8	176.9	-	No	BLSTM-HMM	47.6%	[6]
9	45	57.7 k	No	Guassian Mixture Model (GMM)-HMM, DNN-HMM and TDNN	16.07%	[20]
10	50	-	No	GMM-HMM, Karel's DNN, and TDNN	13.92%	[21]

2.1 Is Tamil a Low Resource Language?

Under-resourced or low-resource language [22] are those languages which may have a few or all of the following aspects; lack of linguistic expertise, less web resources, lack of transcribed speech data, lack of digital phonetical dictionaries, etc. Indeed, Tamil language has lack of linguistic expertise with technological exposure, lack of digital pronunciation dictionaries, extremely limited transcribed speech data, and lack of statistical language models. Tamil language also possess other challenges like many dialects in different regions, code-switching (interchangeably using Tamil and English during conversation), and many non-native/non-fluent speakers.

CoVoST 2 [23], a large-scale multilingual speech-to-text translation (ST) corpus covering translations from 21 languages into English and from English into 15 languages, has used Tamil speech data of only 4 h. A few Tamil speech recognition systems developed recently have utilized very minimal data; 0.5 h [13], 6.5 h [17] and 3 K sentences [16].

2.2 Structure of Tamil Language and Its Challenges

Tamil is the official language of Tamil Nadu (Indian State), Sri Lanka and Singapore, with more than 66 million speakers worldwide. There are numerous recognized geographical dialects and community slangs. Within a single state of India, Tamil Nadu, there are dialects based on different region; Chennai (மெட்ராஸ் பாஷை), Coimbatore (கோவை தமிழ்), Trichy (மத்திய தமிழ்), Madurai (மதுரை தமிழ்), Tirunelveli (திருநெல்வேலி தமிழ்), Kanniyakumari (குமரி தமிழ்). Consider an example sentence "*Friend! When are you coming?",* it is spoken in Tirunelveli dialect as "மக்கா! நீ எப்ப வருத?", in Chennai dialect as "நண்பா! நீ எப்போ வர?" and in literary Tamil as "நண்பனே! நீ எப்பொழுது வருகிறாய்?". It is observed from these examples, that each dialect has their own vocabulary for the same sentence. Hence, it is necessary to create a speech corpus comprising the vocabulary of all dialects to develop a dialect independent ASR. However, addressing this problem is beyond the scope of this paper and we leave this problem open for future researchers.

In Tamil language, Diglossia situation exists in the community [24]. The language has two different versions: Literary and Spoken with significant differences in their structure (sounds) (Fig. 1). For example, in the literary version of words கூடை, நூலை, கடை the sound 'ை' is removed in the spoken version of words கூட, நூல, கட. Tamil language is considered as an intonation language with syllable-based rhythm. The speech features like pitch, loudness, duration, pause, tempo, and rhythm form the intonation system of Tamil language. Tamil graphemic system is generally syllabic in nature and there are a greater number of graphemes than the phonemes. There are instances of a single grapheme used to denote two or more sounds of a single phoneme. The phoneme/k/ has a grapheme <க> which represents [k] in the initial position of a word, as in கண் (eye), [x] in the medial position of a word, as in நகம் (nail) and [g] after homorganic voiced velar nasal [n] as in மாங்காய் (mango) [24]. Tamil language is embodied with such complex structure, semantics, and syntax [25]. Such complexity challenges the development of corpus based on letter-to-sound transcriptions, text to speech engine or other simplistic ways to populate the corpus. Therefore, an end-to-end recognition system is preferable in such a case of low-resourced complex languages. The proposed Tamil ASR is developed using open datasets, web resources to train language model and free computing resources; to demonstrate the cost-effective approach for under-resourced language in a financially constrained environment. We also propose an optimal, cost-effective method to develop a semi-supervised speech corpus using our trained ASR model.

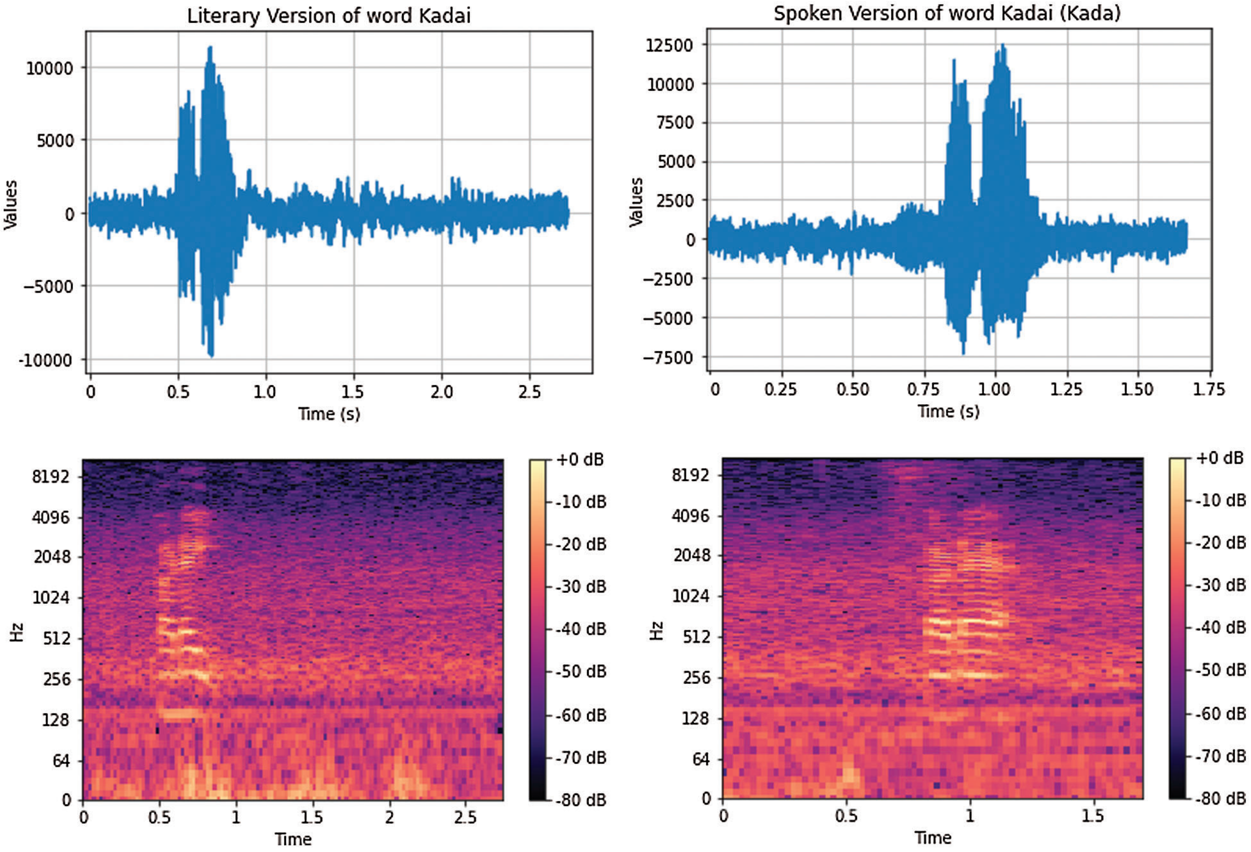


Figure 1: Audio waveforms and corresponding spectrogram of (a) literary and (b) spoken version of word கடை **(Kadai)**

3 Tamil ASR System Architecture

The general pipeline of modelling a speech recognition system is collecting audio utterances, transcribing audio files, mapping letter to sound to form phoneme dictionary, phonetic transcription, acoustic modelling or language modelling, extracting speech features [25] and a learning algorithm with loss minimization. However, the end-to-end speech recognition model does not require phonetic transcription or synchronized audio samples with letter-to-phonemes mapping. Instead, raw audio file is fed into the ASR system, which extracts some audio features, maps the segments of audio to characters, giving recognized characters with its probabilities. In the case of DeepSpeech, the Mel-Frequency Cepstral Coefficient (MFCC) features are extracted from the audio files and fed into a 6-layer deep neural network, obtaining probabilities of entire alphabets. The language model is then used to convert these probabilities into meaningful words and sentences. Connectionist Temporal Classification or CTC is used to minimize loss by ranking the correct transcriptions with higher probabilities. As in Fig. 2, the CTC removes any abnormalities and merges repeated letters, combining a sequence of letters into a valid word.

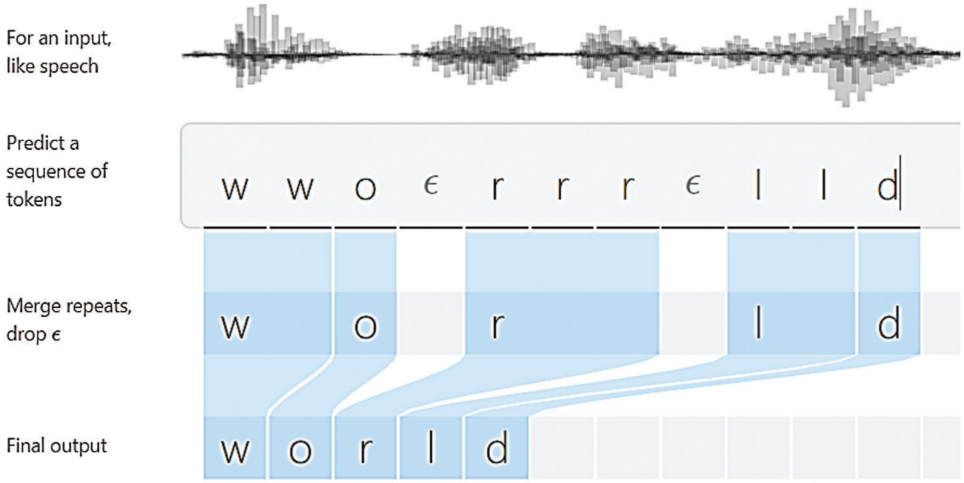


Figure 2: Working of connectionist temporal classification (CTC) [26]. CTC converts the predicted sequence of tokens to valid words with higher probability. In the predicted sequence 'wwo rrr ll d', the pause and the repeated tokens are dropped, giving the final output as 'world' with highest probability

3.1 Speech Corpus

Data collection and processing is the first major task in speech recognition. The speech corpus used for training comprises audio utterances and its corresponding transcriptions. Speech corpus can be of read, conversational or spontaneous speech. To build a read speech corpus, the following minimal steps are to be executed. 1) Prepare large text data (sentences) covering most of the vocabulary of the target language. 2) Record those sentences uttered by a variety of speakers (native, fluent, different age and gender) in different environments (noise, clean, outdoor/indoor). Conversational speech corpus can be built by recording the normal conversation of a speaker on any topic, and then transcribing each of them. Usually, these raw recordings are termed as uncleaned dataset, as post processing must be done to prepare a clean dataset. Post processing steps include preparing the transcriptions of audio utterances, cleaning audio samples (removing too noisy, repeated words, etc.) and synchronization of audio and text transcriptions. The whole process is time consuming and financially expensive (recording setup, incentives to speakers, hosting of data, etc.). The researchers with resource and financial constraints must depend only on publicly available corpus or some innovative crowdsourcing [27].

However, a good open-source initiative to create free speech corpus by Mozilla [28], the Common Voice Corpus 6.1, has recorded speech data of 7300 h covering 60 languages. Common Voice Corpus 6.1 has over 2000 h of recorded speech for English language, while it has only 24 h for Tamil language out of which only 14 h is validated. A generic large vocabulary Tamil speech corpus is yet to be developed and made available to the public. In our paper, we utilize Common Voice and OpenSLR [29] speech corpus for training Tamil ASR. The summary of both the corpora, total words, unique words, total duration, total utterances, average duration of each utterance and average word per utterance is listed in Tab. 2.

3.2 ASR Architecture

The DeepSpeech (v0.7.4) is a modified version of the original TensorFlow implementation of Baidu's End-to-End ASR system [24]. The original version used a bi-directional Recurrent Neural Network (RNN), making it difficult for real-time ASR. In bi-directional RNN, an entire input is fed to get the entire output. However, in the recent v0.7.4, bi-directional layer is replaced with unidirectional layer. This enables the model to process the segments of input data to get partial output, and then pass the same as the initial state for the next segment of input data. The core of the DeepSpeech architecture is a Recurrent Neural Network which is trained on the MFCC extracted from the audio data.

Table 2: Details of common voice and OpenSLR dataset. The 'train' set is used to train the ASR model, while the 'test' set is used to validate the accuracy of the trained ASR model

Dataset	Total words	Total unique words	Total duration (h)	Total utterances	Average duration (sec)	Average word per utterances
Common voice (train)	35598	13857	10.87	9163	4.27	4
Common voice (test)	5753	4434	1.76	1500	4.23	5
OpenSLR	29307	8020	7.05	4277	5.93	8
Common voice (train) + openSLR	64905	20739	17.93	13440	4.80	6

The model consists of six layers. The MFCC is extracted from the raw audio and fed into the input layer which is connected to three fully connected dense layers. Further, it is connected to an unidirectional RNN layer, then to a fully connected dense layer, and finally to an output layer giving estimated probabilities of all the alphabets. Clipped Rectified-Linear unit (ReLu) is used as the activation function. Once the predicted character probabilities for each time segment and characters in the alphabet set are computed, the CTC loss is used to measure the error in prediction. The ASR architecture used in our experiments is illustrated in Fig. 3.

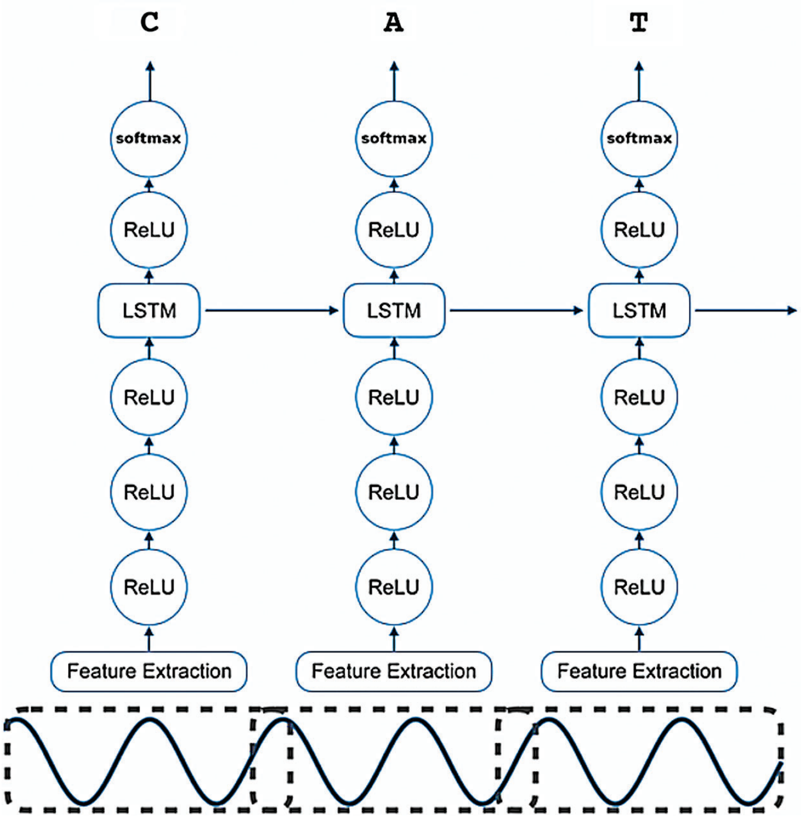


Figure 3: Six-layer DeepSpeech ASR architecture [30]

3.3 Language Model

DeepSpeech outputs an acoustic model, mapping acoustic features to probabilities of alphabets. These probabilities are converted into a sequence of meaningful words using the language model. The language model is trained over a large text corpus of Tamil language, which then assigns probabilities to valid Tamil words and phrases. Once a language model is learnt from the training data, the trained language model assigns higher probability to the valid sentence, than the invalid sentences. Consider an example of sentence with homophone, நான் வெல்லம் சாப்பிட்டேன் (I ate jaggery) and நான் வெள்ளம் சாப்பிட்டேன் (I ate flood). The model will assign higher probability to the former (valid) sentence than the latter (invalid) one, because the word ‘ate’ is most likely to be followed by the word வெல்லம் (jaggery) than the word வெள்ளம் (flood) in the training text corpus, even though both the words are phonetically similar.

A 3-gram language model is trained using kenLM [31], and the training data is a corpus of text data listed in Tab. 3. Four language models are trained using different text data of minimal vocabulary to large vocabulary; to test the speech recognition accuracy using different language models. Language model (LM2) is trained over text data of approximately 6500 news articles [32] and language model (LM3) is trained over 127k Tamil wikipedia articles. The larger language model (LM4) is trained over both news and wikipedia articles. The KenLM scorer built using text data and the account model trained using speech data work together to provide better overall accuracy.

Table 3: Text dataset used for training four different language models These text datasets are scrapped from web resources

Dataset	Total words	Total unique words	Total sentences
CV (train)–LM1	35598	13857	9163
News articles–LM2	130747	34304	13384
Wiki articles–LM3	17670087	1296175	1624478
Wiki & News–LM4	17800834	1330479	1637862

4 Model Training and Results

DeepSpeech (v0.7.4) is used in the experimentation, both for training and testing. The whole experiment is intended to utilize minimal financial resources, hence publicly available free dataset is used as training corpus. CommonVoice Tamil dataset comprising both training set and test set are used for the experimentation. Google Colaboratory (GC) is used as computational resource for training our model. DeepSpeech architecture needs Graphics Processing Unit (GPU) resources to run the training in minimal time, hence GPU is selected as the hardware accelerator in GC. Even though GC has usage time limits while using GPU, checkpoints are saved at regular intervals, which are continued after the time limit is revoked. The specification of GPU in GC is Tesla K80 with 2496 CUDA cores, 12GB VRAM. Typical runtime for training a model takes almost 8 h.

4.1 Training Setup

In DeepSpeech, hyperparameters that are to be chosen and configured during training the model are listed in Tab. 4. The optimal train and test batch size are chosen based on the memory allocation of the GPU, to avoid out-of-memory issues. To utilize deep neural network, 1024 hidden units are configured. Overfitting is avoided by setting the dropout rate at 0.4. Training is performed over two datasets; Common Voice (train) [DS1] and OpenSLR merged with Common Voice (train) [DS2]. A single test set, Common Voice (test), is used for validation of trained models. Character Error Rate (CER) and Word Error Rate (WER) are used as a measure to obtain the efficiency of the trained model. The testing of both the trained models is done using all the four language models, and their results are stated in Tab. 5. To compare the accuracy of proposed models, the test set is also validated using Google's speech-to-text. The Google Speech API processed the whole Common Voice (Test) audio files, with an average WER of 55%, while our proposed model performed better with a best WER of 24.7%. Sample test instances with predicted transcriptions by our proposed model and Google speech-to-text are listed in Tab. 6.

Table 4: Hyperparameter values used for training. Train/Test batch size is chosen based on the GPU specification to avoid out of memory (OOM) errors. Optimal hidden units are selected. Dropout rate is set to avoid overfitting. The training runs for fixed epochs (200) even if it is converged

Parameter	Value
Train/Test batch size	64
Hidden units	1024
Dropout rate	0.4
Learning rate	0.0001
Epoch	200
No early stop	TRUE

Table 5: Performance of ASR model trained on two different dataset is evaluated across four different language models

Training set	Test set	Language model	CER %	WER %
Common voice (Train)	Common voice (Test)	LM1	**11.59**	**24.70**
		LM2	12.45	26.99
		LM3	17.00	37.40
		LM4	16.98	37.38
CV train + OpenSLR	Common voice (Test)	LM1	**11.72**	**25.08**
		LM2	12.46	26.92
		LM3	16.42	36.98
		LM4	16.44	36.97

Table 6: Example recognition results of proposed ASR model on different test instances. Our model has performed better than google speech-to-text API on the given test dataset

Model	WER	Transcripts
Original	-	முந்தஓர் பாட்டுரைத்தாள் அது
DS1-LM1	0.0	முந்தஓர் பாட்டுரைத்தாள் அது
DS1-LM2	0.0	முந்தஓர் பாட்டுரைத்தாள் அது

(Continued)

Table 6 (continued)

Model	WER	Transcripts
DS1-LM3	0.0	முந்தஓர் பாட்டுரைத்தாள் அது
DS1-LM4	0.0	முந்தஓர் பாட்டுரைத்தாள் அது
Google API	1.33	முந்தை ஓர் பாட்டு உரைத்தால் அது
Original	-	இலையிந்த நாட்டினிலே அவனை ஒப்பார்
DS1-LM1	0.25	இலைஎன்ன நாட்டினிலே அவனை ஒப்பார்
DS1-LM2	0.0	இலையிந்த நாட்டினிலே அவனை ஒப்பார்
DS1-LM3	0.5	இலை என்ற நாட்டினிலே அவனை ஒப்பார்
DS1-LM4	0.5	இலை என்ற நாட்டினிலே அவனை ஒப்பார்
Google API	0.5	இலை இந்த நாட்டினிலே அவனை ஒப்பார்
Original	-	திருடனுக்கு அச்சம் தீர்ந்து போயிற்று
DS1-LM1	0.0	திருடனுக்கு அச்சம் தீர்ந்து போயிற்று
DS1-LM2	0.0	திருடனுக்கு அச்சம் தீர்ந்து போயிற்று
DS1-LM3	0.75	தேடலுக்கு அச்சம் தீர்ப்புக்கு
DS1-LM4	0.75	தேடலுக்கு அச்சம் தீர்ப்புக்கு
Google API	0.0	திருடனுக்கு அச்சம் தீர்ந்து போயிற்று

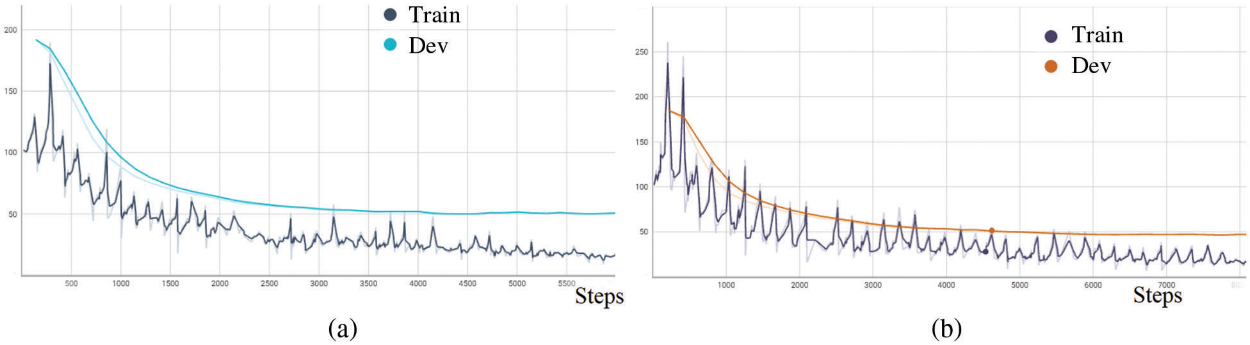
Table 7: Summary of tamil digits corpus built for isolated digit recognition. Speaker 1 (SP1) recorded both training and test instances, while speaker 2 (SP2) and speaker 3 (SP3) recorded only test instances

Numeral	Tamil word	Transliteration	No. of utterances			
			SP1 train	SP1 test	SP2 test	SP3 test
0	சுழியம்	sul̲iyam	50	3	9	4
1	ஒன்று	on̲r̲u	54	5	6	2
2	இரண்டு	iraṇḍu	60	5	8	3
3	மூன்று	mūn̲r̲u	44	6	1	9
4	நான்கு	nān̲ku	54	4	5	5
5	ஐந்து	aindhu	52	3	6	5
6	ஆறு	ār̲u	46	3	4	7
7	ஏழு	ēl̲u	54	3	7	8
8	எட்டு	eṭṭu	48	6	4	7
9	ஒன்பது	on̲pathu	38	2	10	10
			500	50	50	50

The training of the ASR model is performed by splitting the Common Voice (Train) dataset into 80% as train set and 20% as dev set. The model's parameters are tuned to minimize the negative log-likelihood $\sum_{(XY)\in\mathbb{R}} -\log p(\mathrm{Y}|\mathrm{X})$. The training loss converges with the number of iterations; the dev set converges with fewer steps in case of CV dataset, while it takes a few more steps to converge for CV + OpenSLR dataset (Fig. 4). The trained model is saved when the loss is minimum, with standard deviation being less than 0.5 for subsequent 10 iterations, hence avoiding overfitting. The Tab. 6 shows the performance of our trained model (LM1, LM2, LM3, LM4) and Google speech-to-text.

Table 8: WER for isolated digit recognition. Since speaker 2 is non-native, the WER is high, as the model isn't trained with non-native speakers

Speaker	WER %
Speaker 1	7
Speaker 2	23
Speaker 3	5
Average	11.6



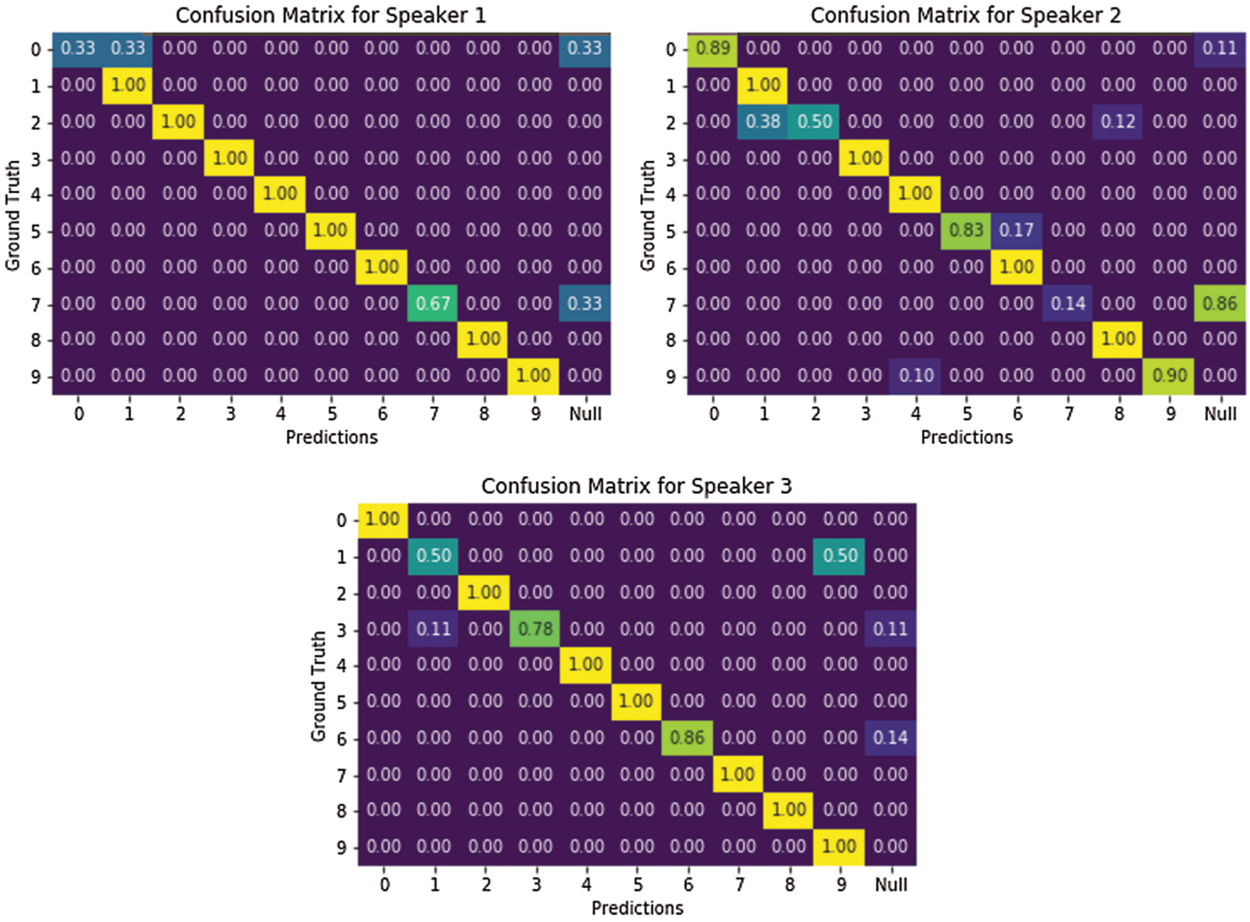
(a)

(b)

Figure 4: Training loss of train and dev set for two different datasets (a) commonVoice (train) and (b) commonVoice (train) + OpenSLR. The former dataset is small compared to the latter dataset; hence the convergence takes more steps for larger dataset

4.2 Transfer Learning for Isolated Tamil Digit Recognition

SpeechStew [33] is an English speech recognition model trained on a massive dataset; combination of seven publicly available datasets totaling to approx. 5140 h. Transfer learning of SpeechStew on CHiME-6, a low resource noisy dataset, produced an improved WER of 38.9 from the official CHiME-6 HMM baseline result of 51.3 WER on dev sets. Such transfer learning on a low resource dataset of 40 H noisy microphone conversational speech recognition is possible only when a massive or numerous publicly available dataset is present. However, for Tamil Language, neither massive dataset nor many public datasets are available, hindering the ASR research. Hence, we have built a pre-trained model using open-source toolkits & open-source dataset and investigated transfer learning approach for limited vocabulary speech recognition. Tamil digits from 0 to 9 are uttered by a single speaker and recorded as a training set (Tab. 7). The digits are displayed randomly one after the other for recording, to avoid sequential utterance of digits. Test set of 50 utterances of digits is recorded by the same speaker and two other speakers. Speakers 1 and 3 are fluent in Tamil language, whereas speaker 2 is non-native. Transfer learning applies a pre-trained model to continue training on a different dataset for a specific use case. CV-LM1 Tamil ASR trained model is used for transfer learning on small digits dataset for isolated Tamil digits recognition. In the pre-trained model, only the last layer is dropped, retaining all the other layers and their weights for transfer learning. The results in Tab. 8 show that the WER obtained in digit recognition is better, while noting that limited dataset is used. Also, the transfer-learned model gives better WER for speaker-independent recognition. The confusion matrix for each speaker is shown in Fig. 5.



Confusion Matrix for Speaker 1

	0	1	2	3	4	5	6	7	8	9	Null
0	0.33	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.33
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Confusion Matrix for Speaker 2

	0	1	2	3	4	5	6	7	8	9	Null
0	0.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.38	0.50	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.83	0.17	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.00	0.86
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.90	0.00

Confusion Matrix for Speaker 3

	0	1	2	3	4	5	6	7	8	9	Null
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.11	0.00	0.78	0.00	0.00	0.00	0.00	0.00	0.00	0.11
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.00	0.00	0.00	0.14
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Figure 5: Confusion matrix of different speaker for isolated digit recognition

4.3 Semi-Supervised Development of Speech Corpus

Large vocabulary corpus is essential to develop a generic speech recognition system. Building such a corpus is time consuming and financially costly. We propose that our pre-trained ASR model could be utilized to create unclean transcription of publicly available audio data of Tamil language. Further manual validation of such transcription could be executed in minimal time, instead of transcribing the audio data from scratch. To demonstrate this, we feed the invalidated Common Voice Tamil dataset to our trained Tamil ASR model and obtain unclean transcriptions. These transcriptions were manually corrected and validated. However, the time taken to validate the unclean transcriptions is 75% lesser than transcribing the audio data from scratch. The results are explained in Tabs. 9 and 10.

Table 9: Performance of our trained ASR model on invalidated common voice dataset. The validated transcript is manually transcribed and the DS1-LM1 transcript is the model predictions. The WER is calculated between the validated and DS1-LM1. Levenshtein distance is the number of insertions, deletions or substitutions needed on the DS1-LM1 transcripts to match the validated transcript. Three different example cases (best, average and worst) are shown

	Transcripts	WER	Levenshtein distance	
Invalidated	கதிபெற வேண்டும் என்றே	0.0	0	Best
Validated	கதிபெற வேண்டும் என்றே			
DS1-LM1	கதிபெற வேண்டும் என்றே			
Invalidated	கொஞ்சம் பறித்துக் கொடுத்தால் உயிர்வாழ்வேன்	0.5	1	Average
Validated	கொஞ்சம் பறித்துக் கொடுத்தால் உயிர்வாழ்வேன்			
DS1-LM1	கொஞ்சம் பறித்துக் கொடுத்தால் உயிர் வாழ்வேன்			
Invalidated	தென்சொல்லெனினும் தமிழ்ச் சொல்லெனினும் ஒக்கும்	1	7	Worst
Validated	தென்சொல்லெனினும் தமிழ்ச் சொல்லெனினும் ஒக்கும்			
DS1-LM1	தென் சொல்லுவினம் தமிழ்ச்சொல் எனினும் ஒக்கும்			

Table 10: Summary of performance of our ASR model on invalidated set of Common Voice dataset (112 utterances). Average WER for different example cases is tabulated

Models prediction	Utterance count	Average WER
Best	44	0.07
Average	36	0.51
Worst	16	1.179
Space insertions	16	1
Total	112	0.506

5 Conclusion

In this paper, we investigated the end-to-end speech recognition system for the low resource language, Tamil and presented the first results of developing a Tamil model using DeepSpeech. Different challenges of low-resourced and semantically complex languages are discussed. We discussed the challenges of developing a corpus for under-resourced languages and presented a novel approach to utilize open source toolkits and datasets. Actively taking part in Mozilla's Common Voice project could lead to development of large corpus with minimal incentivization to volunteers. Our Tamil ASR trained model gave the best result of 24.7% WER, compared to 55% WER by Google speech-to-text. However, our model was evaluated with a limited vocabulary language model. The accuracy of our ASR model depends on the trained language model. This could be improved by building a larger vocabulary language model, assisted by language experts. The proposed model is trained using 14 h of speech corpus, hence the trained ASR model is only suitable for limited vocabulary speech recognition. However, the proposed ASR model could be helpful for the transfer learning approach, which has been demonstrated with simple isolated Tamil digits recognition. Similar approaches could be taken with our trained model for keyword-spotting, isolated word recognition, etc. Use of larger speech corpus (approx. 100 h) to train ASR model will assist in development of generic ASR model. Semi-supervised development of speech corpus using our pre-trained ASR model has demonstrated that large vocabulary corpus could be built with minimal manual assistance. This could further be used to transcribe publicly available audio data and build a large vocabulary corpus. Speech corpus comprising regional dialects needs to be addressed, which could help in the development of a dialect-independent Tamil ASR. The proposed approach can be easily replicated for any other datasets, by utilizing our publicly available Google Colab notebook.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. K. O'Neill, V. Lavrukhin, S. Majumdar, V. Noroozi, Y. Zhang *et al.*, "SPGISpeech: 5,000 h of transcribed financial audio for fully formattedend-to-end speech recognition," submitted to INTERSPEECH, 2021.

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

[3] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg *et al.*, "Deep speech 2: End-to-end speech recognition in English and mandarin," in *Int. Conf. on Machine Learning*, NY, USA, vol. 48, pp. 173–182, 2016.

[4] Y. Zhang, J. Qin, D. S. Park, W. Han, C. C. Chiu *et al.*, "Pushing the limits of semi-supervised learning for automatic speech recognition," arXiv preprint arXiv: 2010.10504, 2020. [Online]. Available: http://arxiv.org/abs/2010.10504.

[5] J. Billa, "ISI ASR system for the Low resource speech recognition challenge for Indian languages," *INTERSPEECH*, Hyderabad, India, pp. 3207–3211, 2018.

[6] C. Liu, Q. Zhang, X. Zhang, K. Singh, Y. Saraf *et al.*, "Multilingual graphemic hybrid ASR with massive data augmentation," in *Proc. of the 1st Joint Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, Marseille, France, pp. 46–52, 2020.

[7] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, pp. 4945–4949, 2016.

[8] Mustaqeem and S. Kwon, "Att-net: Enhanced emotion recognition system using lightweight self-attention module," *Applied Soft Computing*, vol. 102, pp. 107101, 2021.

[9] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong *et al.*, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 28, pp. 802–810, 2015.

[10] Mustaqeem and S. Kwon, "CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network," *Mathematics*, vol. 8, no. 12, pp. 2133, 2020.

[11] Mustaqeem and S. Kwon, "1D-Cnn: Speech emotion recognition system using a stacked network with dilated cnn features," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 4039–4059, 2021.

[12] Mustaqeem and S. Kwon, "A CNN-assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, no. 1, pp. 183, 2020.

[13] Y. Karunanayake, U. Thayasivam and S. Ranathunga, "Sinhala and tamil speech intent identification from English phoneme based ASR," in *Int. Conf. on Asian Language Processing (IALP)*, Shanghai, China, pp. 234–239, 2019.

[14] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara and S. Watanabe, "Transfer learning of language-independent end-to-end ASR with language model fusion," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, pp. 6096–6100, 2019.

[15] Y. Chen, J. Y. Hsu, C. K. Lee and H. Y. Lee, "DARTS-Asr: Differentiable architecture search for multilingual speech recognition and adaptation," *INTERSPEECH*, Shanghai, China, pp. 1803–1807, 2020.

[16] S. Lokesh, P. M. Kumar, M. R. Devi, P. Parthasarathy and C. Gokulnath, “An automatic tamil speech recognition system by using bidirectional recurrent neural network with self-organizing map,” *Neural Computing and Applications*, vol. 31, no. 5, pp. 1521–1531, 2019.

[17] A. Madhavaraj and A. G. Ramakrishnan, “Design and development of a large vocabulary, continuous speech recognition system for tamil,” in *IEEE India Council Int. Conf. (INDICON)*, Uttarakhand, India, pp. 1–5, 2017.

[18] A. Madhavaraj, H. R. S. Kumar and A. G. Rarnakrishnan, “Online speech translation system for tamil,” *INTERSPEECH*, Hyderabad, India, pp. 1966–1967, 2018.

[19] B. Pulugundla, M. K. Baskar, S. Kesiraju, E. Egorova, M. Karafiát *et al.*, “BUT system for Low resource Indian language ASR,” *INTERSPEECH*, Hyderabad, India, pp. 3182–3186, 2018.

[20] N. Fathima, T. Patel, C. Mahima and A. Iyengar. “TDNN-Based multilingual speech recognition system for Low resource Indian languages,” *INTERSPEECH*, Hyderabad, India, pp. 3197–3201, 2018.

[21] B. M. L. Srivastava, S. Sitaram, R. K. Mehta, K. D. Mohan, P. Matani *et al.*, “Interspeech 2018 Low resource automatic speech recognition challenge for Indian languages,” in *6th Int. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2018)*, Gurugram, India, pp. 11–14, 2018.

[22] L. Besacier, E. Barnard, A. Karpov and T. Schultz. “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication*, vol. 56, pp. 85–100, 2014.

[23] C. Wang, A. Wu and J. Pino, “Covost 2: A massively multilingual speech-to-text translation corpus,” arXiv preprint 2007.10310, 2020.

[24] “Tamil language variations,” Central Institute of Indian Languages, 2021. [Online]. Available: http://lisindia.ciil.org/Tamil/Tamil_vari.html.

[25] S. A. Mahar, M. H. Mahar, J. A. Mahar, M. Masud, M. Ahmad *et al.*, “Superposition of functional contours based prosodic feature extraction,” *Intelligent Automation and Soft Computing*, vol. 29, no. 1, pp. 183–197, 2021.

[26] A. Hannun, “Sequence modelling with CTC,” in *Distill*, Distill Working Group, San Francisco CA, USA, 2017.

[27] A. A. Raza, A. Athar, S. Randhawa, Z. Tariq, M. B. Saleem *et al.*, “Rapid collection of spontaneous speech corpora using telephonic community forums,” *INTERSPEECH*, Hyderabad, India, pp. 1021–1025, 2018.

[28] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler *et al.*, “Common voice: A massively-multilingual speech corpus,” arXiv preprint arXiv: 1912.06670, 2019.

[29] F. He, S. H. C. Chu, O. Kjartansson, C. Rivera, A. Katanova *et al.*, “Open-source multi-speaker speech corpora for building gujarati, kannada, malayalam, marathi, tamil and telugu speech synthesis systems,” in *The 12th Language Resources and Evaluation Conf.*, Palais du Pharo, Marseille, pp. 6494–6503, 2020.

[30] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos *et al.*, “Deep speech: Scaling up end-to-end speech recognition,” arXiv preprint arXiv: 1412.5567, 2014.

[31] K. Heafield, “KenLM: Faster and smaller language model queries,” in *Proc. of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, pp. 187–197, 2011.

[32] G. Arora, “iNLTK: Natural language toolkit for indic languages,” in *Proc. of Second Workshop for NLP Open Source Software (NLPOSS)*, pp. 66–71, 2020.

[33] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le *et al.*, “Speechstew: Simply mix all available speech recognition data to train one large neural network,” *Workshop on Machine Learning in Speech and Language Processing (Online)*, Brno, Czechia, 2021.