Applicability of End-to-End Deep Neural Architecture to Sinhala Speech Recognition

Buddhi Gamage[1*], Randil Pushpananda[2], Thilini Nadungodage[2], Ruvan Weerasinghe[2]
[1]Faculty of Computing, University of Sri Jayewardenepura
[2]Language Technology Research Laboratory, University of Colombo School of Computing
bgamage@sjp.ac.lk, rpn@ucsc.cmb.ac.lk, hnd@ucsc.cmb.ac.lk, arw@ucsc.cmb.ac.lk

Abstract—**This research presents a study on the application of end-to-end deep learning models for Automatic Speech Recognition in the Sinhala language, which is characterized by its high inflection and limited resources. We explore two e2e architectures, namely the e2e Lattice-Free Maximum Mutual Information model and the Recurrent Neural Network model, using a restricted dataset. Statistical models with 40 hours of training data are established as baselines for evaluation. Our pretrained end-to-end Automatic Speech Recognition models achieved a Word Error Rate of 23.38% by far the best word-error-rate achieved for low resourced Sinhala Language. Our models demonstrate greater contextual independence and faster processing, making them more suitable for general-purpose speech-to-text translation in Sinhala.**

Index Terms—**Speech Recognition, Deep Learning, Transfer learning**

I. Introduction

The field of Automatic Speech Recognition (ASR) encompasses two main architectures for training ASR systems: the Statistical ASR architecture and the End-to-End (e2e) Deep Neural architecture. While Statistical ASR remained state of the art for many years, the landscape shifted towards e2e ASR systems after 2015 due to their superior performance. The key distinction between these architectures lies in the number of models required for training the ASR system. Statistical ASR relies on three distinct models: acoustic models, pronunciation models, and language models. In contrast, e2e ASR compresses these three models into a single Deep Neural Network (DNN) [1]. Given the rising popularity of e2e architecture in Natural Language Processing (NLP) and speech recognition, numerous studies have explored its application in developing ASR systems for various languages. Previous research focusing on English speech recognition has demonstrated improved outcomes when utilizing the e2e architecture compared to traditional statistical approaches [2].

In the realm of ASR system development for the Sinhala language , prior and ongoing research has predominantly focused on utilizing statistical ASR architecture , Gaussian Mixture Model with Hidden Markov Model (GMM -HMM) based models, and Hybrid Deep-Neural-Network with Hidden Markov Model (DNN-HMM) based models [3], [4]. However, the adoption of the e2e architecture for Sinhala ASR repre-sents a novel approach that holds the potential to enhance available resources . Specifically , the e2e architecture facilitates transfer learning , a trending technique in low -resource speech recognition, which can effectively improve accuracy [5]. In the research community , several tools are available for creating ASR systems, such as Kaldi, DeepSpeech, Espresso, and Wav2letter. However, there has been no comparative analysis of these tools specifically for the Sinhala language . Hence , it is imperative to determine the most suitable approach for developing ASR systems in Sinhala by leveraging the various available tools.

This paper focuses on conducting a comprehensive study on e2e Deep Neural Network (DNN) architecture -based ASR systems for Sinhala speech recognition . Specifically , we explore two e2e models, namely the e2e LF-MMI model and the RNN model and with Pretraining, which are created using three distinct toolkits : Kaldi , Espresso , and DeepSpeech . The performance of each e2e model will be rigorously evaluated and compared against established statistical models , including GMM-HMM, DNN-HMM, and combinational models such as SGMM-DNN. By systematically analyzing and contrasting the outcomes of these models, we aim to gain valuable insights into the efficacy of e2e DNN architectures for Sinhala speech recognition.

The structure of this paper is organized as follows. Section 2 provides an overview of the related studies in the field of ASR, highlighting the existing research on e2e architectures and their applications in various languages. In Section 3, we delve into the methodology employed for this study , including data preparation and implementation of the e2e DNN models using different toolkits . This section offers a comprehensive description of the experimental setup and procedures. Section 4 presents the results obtained from the evaluation of the e2e models and their comparison with statistical models, presenting an in-depth analysis of the performance metrics. Furthermore , Section 5 outlines the conclusions drawn from the study and proposes potential avenues for future research and improvement in the field of Sinhala speech recognition using e2e DNN architectures.

Correspondence : Buddhi Gamage (e-mail: bgamage @sjp.ac.lk)
Received: 08-01-2024 Revised: 18-03-2024 Accepted: 20-03-2024

Buddhi Gamage is from Faculty of Computing, University of Sri Jayewardenepura (bgamage @sjp.ac.lk), Randil Pushpananda , Thilini Nadungodage and Ruvan Weerasinghe are from Language Technology Research Laboratory, University of Colombo School of Computing.(rpn@ucsc.cmb.ac.lk, hnd@ucsc.cmb.ac.lk, arw@ucsc.cmb.ac.lk)

DOI: https://doi.org/10.4038/icter.v17i1.7273

II. Literature Review

The field of speech recognition has a rich history that dates back to the early 1920s [6]. Over the years, significant advancements have been made in various approaches and models. Initially, template-based methods were used, but in the 1980s, statistical modelling approaches, particularly Hidden Markov Models (HMM), gained prominence and replaced the earlier methods [7]. The introduction of Deep Neural Networks (DNN) in the era of deep learning revolutionized ASR, leading to improved acoustic models and surpassing the performance of traditional HMM-based models. However, HMM-based models still dominate the field due to their practicality and the challenges associated with training and decoding processes [1].

Recently, there has been a shift towards end-to-end (e2e) architectures in ASR. Unlike HMM-based models, e2e models directly map input audio sequences to word or character sequences, eliminating the need for intermediate states and streamlining the overall ASR process. However, e2e models require a large amount of speech data for higher recognition accuracy, making them less suitable for low-resource scenarios [1].

To address the low-resource challenge, transfer learning and unsupervised learning techniques have gained popularity. Researchers have explored the use of transfer learning techniques, such as weight transfer and multitask learning, to tackle the scarcity of training data. Additionally, advancements in meta-learning and unsupervised pre-training techniques have shown promising results in improving ASR accuracy for low-resource languages [8].

Several research papers published in 2020 have focused on transfer learning and low-resource speech recognition, utilizing models such as LF-MMI and Deepspeech. These papers have contributed valuable insights and advancements to the field [5], [9]–[11].

In summary, the current focus of speech recognition research is on addressing the low-resource problem through transfer learning techniques. The utilization of e2e LF-MMI models and advancements in deep learning offer promising opportunities for improved ASR performance and efficiency in both high-resource and low-resource scenarios.

III. Methodology

A. Approach

The proposed solution for evaluating the applicability of e2e deep neural architecture is depicted in Fig. 1. In this study, we investigate two e2e models extensively, which are considered key components of the research. These models are carefully analyzed and evaluated to assess their effectiveness in the context of our study.

1) Recurent Neural Network (RNN) model: For training the RNN models, we employed the default 6-layer neural network architecture, with the recurrent layer placed in the 4th layer. Each hidden layer consisted of 375 hidden units. Throughout our study, we utilized the default RNN architecture provided in Deepspeech, as described in [12]. However, since our focus was on the Sinhala language, we needed to modify the alphabet used in the models. In addition to the characters from the Sinhala Unicode character table, we included zero-width space, zero-width joiner, and zero-width non-joiner characters in the alphabet. This inclusion was necessary to address and mitigate any errors that may arise during training specifically for the Sinhala language.

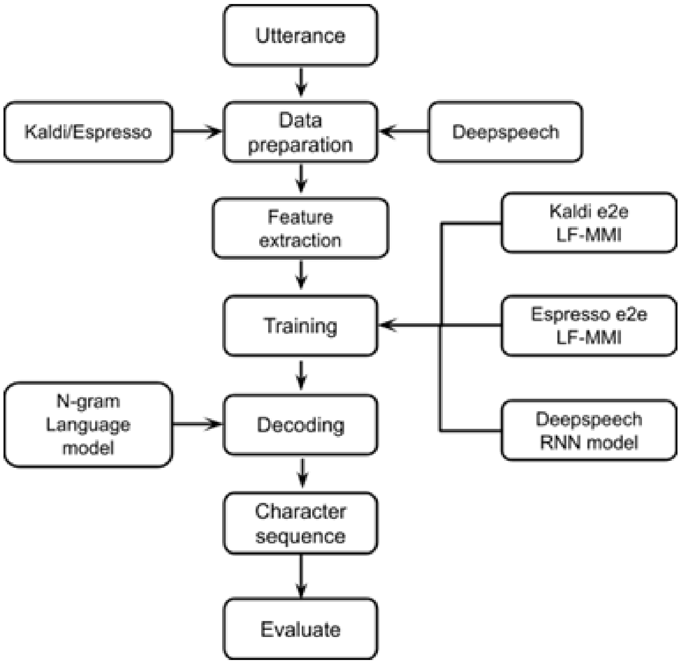

Fig. 1. High-level architecture of the research design

2) E2e Lattice-Free Maximum Mutual Information (e2e LF-MMI) model: To train the WSJ dataset, we employed the default Neural Network (NN) architecture specified in the training recipes. For the Kaldi e2e LF-MMI model, we utilized Factored Time Delay Neural Networks (TDNNf) as per the standard Kaldi WSJ recipe. This neural network is comprised of 13 TDNNf layers along with a rank reduction layer. The TDNNf layer consisted of 1024 units and 128 bottleneck units. We followed the default hyperparameters outlined in the standard recipe, with 10 and 30 epochs [13].

In the Espresso e2e LF-MMI models, we adopted the TDDN+LSTM network specifications. This architecture included 7 TDNN layers and 3 LSTM layers. The TDNN layers had a dimensionality of 512, while the LSTM layers comprised 128 recurrent and non-recurrent dimensions. For further details regarding the e2e LF-MMI models and the default NN architecture, please refer to [13].

B. Data Preparation

Data preparation is a crucial step in the ASR pipeline, as the reliability and accuracy of the ASR system heavily rely on the consistency and integrity of this stage [3]. In our study, we utilized three toolkits, but data preparation was only required for Kaldi and Deepspeech. Interestingly, Espresso, as a system, does not necessitate any additional data preparation. Instead, we could utilize the same prepared files from Kaldi for training models through Espresso as well [14].

When preparing the data for Deepspeech, it was essential to ensure that each audio file contained a single utterance. Additionally, the training process of Kaldi e2e LF-MMI did not support the use of a segment file, which contains the length of each utterance in a single audio file. Therefore, special consideration was given to these requirements during the data preparation phase. In this study, we utilized recordings collected from UCSC LTRL (University of Colombo School of Computing Language Technology Research Laboratory), which provided us with a dataset comprising 40 hours of training data. These recordings were gathered using Praat and Redstart tools, which are commonly used in speech research and analysis. The dataset from UCSC LTRL served as the primary source of training data for our experiments in building e2e ASR systems for Sinhala speech recognition.

1) Dataset: The training process involved a total dataset consisting of recordings from 113 speakers, with 79 female speakers and 34 male speakers. Within the training dataset, there were 67 female speakers and 27 male speakers, resulting in a total of 17,848 sentences. This amounted to approximately 25 hours of speech data. For fine-tuning the models, a validation dataset of 2,002 speech utterances was used, which included recordings from 8 female speakers and 3 male speakers. During the testing phase, a separate dataset was utilized, consisting of recordings from 4 female speakers and 4 male speakers. In total, this dataset included 80 speech sentences. The training process was conducted at a sample rate of 16kHz, and further details can be found in [3]. Table I provides an overview of the overall details regarding the datasets used in this study.

TABLE I
DETAILS OF TRAIN, VALIDATION, AND TEST DATA SETS

Dataset	Male	Female	Utterances
Train	27	67	17848
Dev	3	8	2002
Test	4	4	80

2) Lexicon: The lexicon plays a crucial role in the pronunciation model of a statistical ASR system, as it maps words to their corresponding spoken phone sequences [3]. In this study, the lexicon was created using two tools: ”Sinhala G2P Conversion” [15] and ”Subasa Transliterator”. These tools were employed to generate the necessary mappings between words and their respective phonetic representations. For further information and specific details, please refer to [3].

3) Corpus: Three corpora were utilized in this project: the UCSC Novel Corpus consisting of 90,000 unique sentences, the Chatbot Corpus with 388 unique sentences, and a corpus created using the active learning method comprising 20,000 unique sentences. These corpora were combined to form a new corpus for the generation of n-gram language models. The summary statistics of the corpus can be found in Table II. Two toolkits were employed to create the n-gram language models: SRILM [16] and KenLM [17]. Perplexity calculations were performed on the testing dataset, and a 4-gram language model was selected based on the study. Additional details regarding the Language Models can be found in Table III.

TABLE II
CORPUS STATISTICS

Vocabulary Size	243339
Total number of Sentences	119621
Total number of words	119494

TABLE III
PERPLEXITIES OF LANGUAGE MODELS

Language Model	Perplexity
Witten-Bell 3grams	9.393376
Witten-Bell 4grams	8.108833

C. Baseline Models

In this study, 2 baseline models were considered, excluding basic statistical models monophone and triphone models. These models, as described in [18], are:

- Hybrid System (Dan’s DNN)
- E2e LF-MMI Model

The process of creating these baseline models is detailed in [18]. A total of 40 hours of data were used for training. For feature extraction, Mel Frequency Cepstral Coefficients (MFCC) were computed using 13 coefficients, including the zero-order coefficient. The features were extracted every 10ms with a 25ms Hamming window, following the standard measurement mentioned in [19]. The results obtained from the baseline models are presented in Table IV.

D. LF-MMI Model

For creating the e2e models, we opted for phone-based training. Unlike Deepspeech, the Kaldi and Espresso toolkits do not utilize an alphabet. Instead, we can rely on the lexicon to map words to phone sequences during the decoding process. The architecture used to create the LF-MMI models is depicted in Fig. 2.

In the e2e models, we extract 40-dimensional MFCC features from 25ms frames every 10ms, following the default setting used in the WSJ recipe [13]. We apply zero mean and unit variance normalization on a per-speaker basis, without any additional feature normalization or transformation. Unlike the baseline models, we do not perform re-alignments during the training process.

Data augmentation is carried out using 2-fold speed perturbation in all experiments. This perturbation modifies the length of each utterance to one of the distinct lengths, ensuring that no padding with silence is required [13].

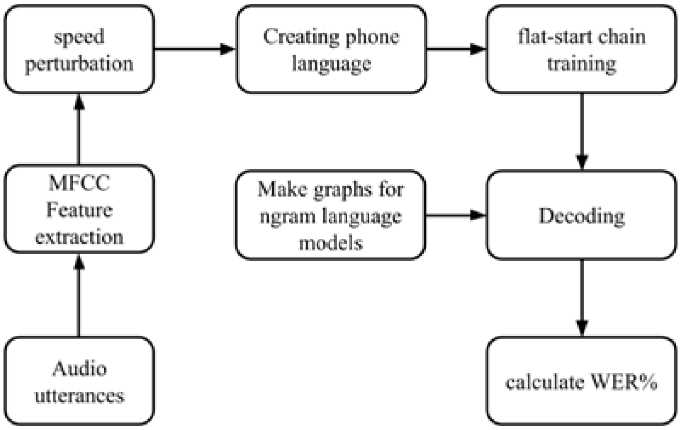

Fig. 2. e2e LF-MMI model Architecture

Unlike in traditional statistical ASR, e2e ASR decodes utterances into character sequences. Therefore, we need a phone language for the denominator graph during the decoding process. We initiate the training of models in both the Kaldi and Espresso toolkits using the aforementioned NN settings. Espresso employs an updated version for creating numerator graphs.

To train the e2e models, we can utilize a different lang directory that includes information about the desired n-gram language models, along with a compatible wordlist and language model. The mkgraph.sh script is used to train the e2e models using such language models.

E. RNN Model

In contrast to other approaches, Deepspeech does not utilize phones to train models [12]. Instead, as mentioned earlier, it employs an alphabet specific to the training language to generate character sequences using a large DNN. Additionally, Deepspeech provides the flexibility to use a separate n-gram language model for decoding utterances. This is referred to as an External Scorer in the Deepspeech documentation.

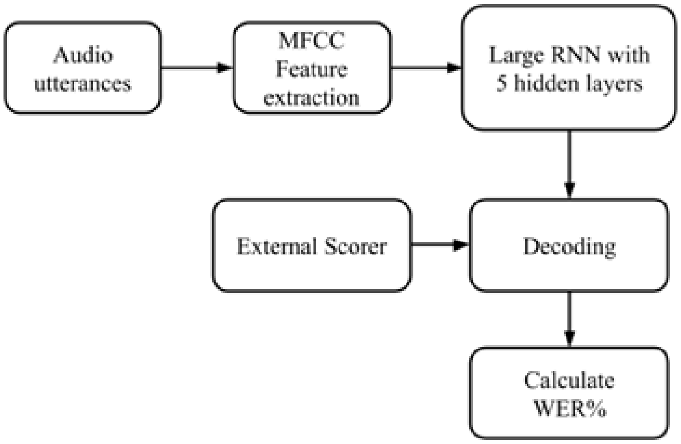

Fig. 3. RNN Model Architecture

In the Deepspeech architecture, 26 MFCC features are extracted, which is the standard setting for a 16kHz sample rate [12]. These features are passed through the first three non-recurrent layers, which utilize the Rectified-Linear (Relu) activation function. The fourth layer is a recurrent layer that includes hidden units with forward recurrence. The fifth layer is another non-recurrent layer that takes the forward units as inputs. The output layer predicts the probabilities of characters for each time slice. To improve the accuracy of the output, an external scorer can be created and used. The Word Error Rate (WER) is then calculated using the testing dataset. The basic structure of the Deepspeech model is depicted in Figure 3, and models are trained for different numbers of epochs, such as 30, 50, and 100.

After training, an ”output_graph.pb” model file is generated. However, loading this model into memory during inference can lead to increased loading time and memory consumption. To mitigate this, TensorFlow provides tools that allow data to be read directly from disk, avoiding the need to load the entire model into memory.

F. Pretrained Model

In order to overcome the limitations posed by the dataset, transfer learning has proven to be a successful approach to developing an ASR system [20]. In this study, we employed the v0.9.3 English pre-trained model as the source model and replaced the output layer, which consisted of the English alphabet, with the Sinhala alphabet. This was done because the specific output layer of the source model is not crucial for our purposes [21].

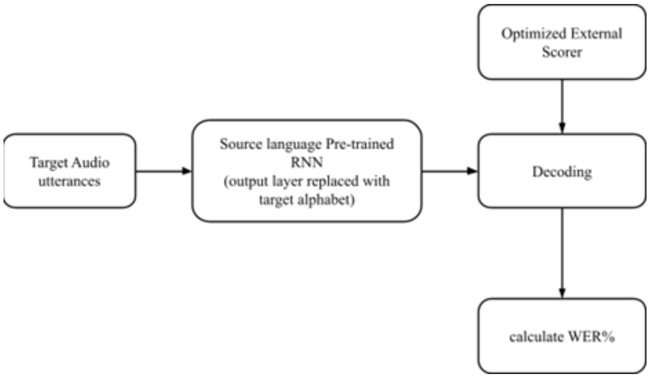
Optimized External Scorer
Target Audio utterances
Source language Pre-trained RNN (output layer replaced with target alphabet)
Decoding
calculate WER%

Fig. 4. Pretrained Model Architecture

IV. Results And Discussion

The training and decoding processes for all deep neural architectures, as well as the evaluation of the models, were performed on a server equipped with 4 GPUs - GeForce RTX 2080 Ti, each with a capacity of 10.8GB. During training, all 4 GPUs were utilized, leveraging CUDA to accelerate the deep learning training process.

The performance of the e2e Sinhala ASR systems was evaluated in terms of accuracy, specifically on recordings captured in noisy environments. The accuracy can be measured

TABLE IV
WER OF ALL TRAINED MODELS OVER THE TESTING DATA AND EXTERNAL LANGUAGE MODELS USED FOR

Model	Tool	WERs
Baseline **- Hybrid System (Dan's DNN** **- E2e LF-MMI Model**	Kaldi	27.79 28.55
E2e LF-MMI Model	Espresso	30.21
RNN Model	DeepSpeech	43.80
Pretrained Model	DeepSpeech	23.38

by calculating either the Word Error Rate (WER) or the Sentence Error Rate (SER). WER represents the number of incorrectly identified words out of the total number of words in the audio sample used for recognition, while SER represents the number of improperly identified sentences out of the total number of sentences. Throughout the study, WER was used as the primary evaluation metric [4].

A. Baseline models

The baseline models included the hybrid DNN model and e2e LF-MMI model which were built on top of the alignments obtained from the LDA+MLLT+SAT (tri3) triphone model. In study [18], the e2e LF-MMI model achieved a WER of 28.55% and the hybrid DNN model achieved 27.79% WER with only 40 hours of training data. Table IV) presents a comparison of the results obtained in this study with the same architecture.

B. E2e Model

The results for the e2e LF-MMI models, both in Kaldi and Espresso, are presented in Table IV. In Kaldi, the model achieved a WER of 28.55% with 10 epochs of training. It is worth noting that when training with GPUs in Kaldi, the available 4 GPUs are treated as a single GPU in exclusive mode, allowing for higher frames during training. For the 10-epoch training, 3 million frames were used per iteration. On the other hand, in Espresso, the highest number of frames that could be used per iteration was 0.12 million, resulting in a WER of 30.21

Comparing the two models, it can be observed that Kaldi achieved a higher accuracy with a 1.66% lower WER compared to Espresso. However, it is important to consider that the performance of the models can be influenced by various factors such as training data, model architecture, and hyperparameters.

As a fully e2e model, the RNN does not use HMM and instead operates at the character level using an alphabet. The (Table 6) displays the Word Error Rate (WER) obtained for the system.

The current WER achieved in the RNN model is 43.80%. Upon evaluating the sentences and examining the outputs, we can identify areas that require improvement and potential augmentations to enhance the performance of future studies, such as transfer learning.

By employing transfer learning techniques and utilizing an English pre-trained model, the RNN-based ASR system achieved significant reduction in Word Error Rate (WER). The WER decreased from 43.80% to 23.38%, resulting in a remarkable improvement of 20.42%. This achievement represents the best WER recorded for the low-resource Sinhala Language to date.

To further enhance the ASR system 's performance , additional improvements can be made . One approach is to incorporate more Sinhala speech data into the training process. Additionally , considering the similarities within the Indo-Aryan Language family , utilizing speech data from other languages within this family could also contribute to reducing the WER even further . By leveraging these strategies , it is possible to continue advancing the accuracy and effectiveness of the ASR system for the Sinhala Language.

By analyzing the errors and discrepancies in the model 's outputs for the selected sentences, we can gain insights into the specific challenges faced by the model and take steps to address them. These improvements may involve refining the training data, applying data augmentation techniques , optimiz - ing model architecture and hyperparameters , or incorporating advanced techniques like transfer learning . By iteratively refining the model based on these observations , we can strive to achieve better accuracy in future studies.

C. Evaluation

Three sentences were randomly selected from different individuals. The recordings were conducted in their respective environments using their own equipment, with a sample rate of 44.1 Hz. The baseline model chosen for comparison was Hybrid Dan's model, as it achieved the lowest WER among the baseline models. Additionally, the accurate e2e models created in Kaldi, Espresso, and Deepspeech were used for evaluation.

shown in [3], [18] that the current Sinhala ASR system performs well in the context of news and number readings due to the training data primarily focusing on these areas. However, they are underperformed in normal day-to-day conversations.

In this study , it was observed that the statistical models employed in the research had a tendency to misidentify utterances during normal day-to-day conversations . However, the pre-trained models showed higher accuracies compared to other models. This discrepancy in accuracy can be attributed to the context -dependent decoding of statistical models , which often leads to lower accuracy for sentences with fewer words.

In contrast , the e2e models demonstrated a more context-independent nature, despite being trained on data that exhibits context dependency . As a result , e2e techniques prove to be more suitable for the development of a general Automatic Speech Recognition (ASR) system . These models are able to handle a wide range of speech inputs, including both longer and shorter sentences , and exhibit a higher level of accuracy compared to traditional statistical models.

D. Limitations

The findings of this study should be interpreted considering certain limitations. it is important to note that this result may be biased towards the specific testing dataset used in this study. While the proposed model has by far the best WER 23.38%

it still falls short of achieving the state-of-the-art WER. This indicates that the RNN model would benefit from a larger amount of speech data to achieve higher levels of accuracy. Therefore, the proposed model cannot be considered the best option given the available data.

Additionally, it is worth mentioning that the training process was conducted on a server equipped with 4 RTX 2080TI GPUs. The training duration for each model was 3 days, which presented challenges in fine-tuning the models. Due to time constraints, only the language parameters (alpha and beta) were fine-tuned, and further training using the augmentations of WAV files was not considered in this research. This limitation suggests that additional improvements could be achieved by incorporating more comprehensive fine-tuning techniques and leveraging data augmentation methods, although these would require significant additional time and resources.

V. Conclusion

In this research, we successfully achieved our objectives by developing a context-independent and faster model for Sinhala speech recognition, specifically for general-purpose speech-to-text transcription, using an end-to-end (e2e) approach. This research aligns with existing literature, which emphasizes the suitability of DNN approaches for ASR systems.

Currently, our e2e pre-trained model implemented on the DeepSpeech toolkit achieves a Word Error Rate (WER) of 23.38% for Sinhala speech recognition. However further improvements can be made through fine-tuning and utilizing advanced techniques.

The field of speech recognition is evolving to address the challenge of low-resource languages. Extensive datasets are available for languages such as English and French, accompanied by state-of-the-art results. To overcome the low-resource problem, a common solution involves transfer learning from high-resource languages to low-resource languages. Deepspeech provides scripts for transfer learning using Common Voice data for the English language, which offers 2,181 hours of training data. In the e2e LF-MMI technique, transfer learning can be achieved through weight transfer and multi-task training [9]. Hence, based on the results of this study, it would be beneficial to explore data augmentation techniques and optimize parameters for the aforementioned transfer learning techniques.

In conclusion, our research highlights the effectiveness of the e2e approach for Sinhala speech recognition and the superiority of DNN-based models. We have achieved promising results, and future work should focus on refining the model through fine-tuning and exploring transfer learning techniques to address the low-resource challenge and further improve the accuracy of Sinhala speech recognition systems.

VI. Conflict of interest statement

The authors declare that they have no financial or other substantive conflicts of interest that could be perceived as influencing the results or interpretation of their research. This research project received no specific financial support from any external sources.

VII. Acknowledgement

We would like to express our sincere gratitude to all individuals and organizations who have contributed to the successful completion of this research project. We extend our appreciation to the participants who generously provided their speech data, our supervisors and advisors for their guidance, the research team members for their assistance in data collection, the technical support team for their expertise, and the academic institutions and research centres for providing necessary resources. Lastly, we thank our colleagues, friends, and family members for their encouragement and support throughout this research journey.

References

[1] Dong Wang, Xiaodong Wang, and Shaohe Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018, 2019.

[2] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[3] Buddhi Gamage, Randil Pushpananda, Ruvan Weerasinghe, and Thilini Nadungodage. Usage of combinational acoustic models (dnn-hmm and sgmm) and identifying the impact of language models in sinhala speech recognition. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 17–22. IEEE, 2020.

[4] Hirunika Karunathilaka, Viraj Welgama, Thilini Nadungodage, and Ruvan Weerasinghe. Low-resource sinhala speech recognition using deep learning. In *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 196–201. IEEE, 2020.

[5] Mihaela C Stoian, Sameer Bansal, and Sharon Goldwater. Analyzing asr pretraining for low-resource speech-to-text translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913. IEEE, 2020.

[6] Shipra J Arora and Rishi Pal Singh. Automatic speech recognition: a review. *International Journal of Computer Applications*, 60(9), 2012.

[7] Kai-Fu Lee. On large-vocabulary speaker-independent continuous speech recognition. *Speech communication*, 7(4):375–379, 1988.

[8] Vladimir Bataev, Maxim Korenevsky, Ivan Medennikov, and Alexander Zatvornitskiy. Exploring end-to-end techniques for low-resource speech recognition. In *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*, pages 32–41. Springer, 2018.

[9] Pegah Ghahremani, Vimal Manohar, Hossein Hadian, Daniel Povey, and Sanjeev Khudanpur. Investigation of transfer learning for asr using lf-mmi trained neural networks. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 279–286. IEEE, 2017.

[10] Aashish Agarwal and Torsten Zesch. Ltl-ude at low-resource speech-to-text shared task: Investigating mozilla deepspeech in a low-resource setting. *SwissText/KONVENS*, 31:40–47, 2020.

[11] Yonas Woldemariam. Transfer learning for less-resourced semitic languages speech recognition: the case of amharic. In *Proceedings of the 1st joint workshop on spoken language technologies for under-resourced languages (SLTU) and collaboration and computing for under-resourced languages (CCURL)*, pages 61–69, 2020.

[12] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[13] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur. End-to-end speech recognition using lattice-free mmi. In *Interspeech*, pages 12–16, 2018.

[14] Yiming Wang, Tongfei Chen, Hainan Xu, Shuoyang Ding, Hang Lv, Yiwen Shao, Nanyun Peng, Lei Xie, Shinji Watanabe, and Sanjeev Khudanpur. Espresso: A fast end-to-end neural speech recognition toolkit. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 136–143. IEEE, 2019.

[15] Thilini Nadungodage, Chamila Liyanage, Amathri Prerera, Randil Pushpananda, and Ruvan Weerasinghe. Sinhala g2p conversion for speech processing. In *SLTU*, pages 112–116, 2018.

[16] Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*, 2002.

[17] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197, 2011.

[18] Buddhi Gamage, Randil Pushpananda, Thilini Nadungodage, and Ruwan Weerasinghe. Improve sinhala speech recognition through e2e lf-mmi model. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 213–219, 2021.

[19] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[20] Julius Kunze, 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] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.