Large Scale Speech Recognition for Low Resource Language Amharic, an End-to-End Approach

Yohannes Ayana Ejigu * and Tesfa Tegegne Asfaw

Posted Date: 15 February 2024

doi: 10.20944/preprints202402.0813.v1

Keywords: Automatic speech recognition; Convolutional Neural Network; Connectionist Temporal Classification; End-to-End; Neural network; Erosion; Recurrent Neural Network

 doi:10.20944/preprints202402.0813.v1

Article

Large Scale Speech Recognition for Low Resource Language Amharic, An End-to-End Approach

Yohannes Ayana Ejigu [1,*] and Tesfa Tegegne Asfaw [2]

[1] Department of Artificial Intelligence and Data Science, Faculty of Computing, Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia
[2] Department of Computer science, Faculty of Computing, Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia
* Correspondence: yohannesayana10@gmail.com

Abstract: Speech recognition, or automatic speech recognition (ASR), is a technology designed to convert spoken language into text using software. However, conventional ASR methods involve several distinct components, including language, acoustic, and pronunciation models with dictionaries. This modular approach can be time-consuming and may influence performance. In this study, we propose a method that streamlines the speech recognition process by incorporating a unified recurrent neural network (RNN) architecture. Our architecture integrates a convolutional neural network (CNN) with an RNN and employs a connectionist temporal classification (CTC) loss function. Key experiments were carried out using a dataset comprising 576,656 valid sentences, using erosion techniques. Evaluation of the model performance, measured by the word error rate (WER) metric, demonstrated remarkable results, achieving a WER of 2%. This approach has significant implications for the realm of speech recognition, as it alleviates the need for labor-intensive dictionary creation, enhancing the efficiency and accuracy of ASR systems, and making them more applicable to real-world scenarios. For future enhancements, we recommend the inclusion of dialectal and spontaneous data in the dataset to broaden the model's adaptability. Additionally, fine-tuning the model for specific tasks can optimize its performance for targeted objectives or domains, further enhancing its effectiveness in those areas.

Keywords: automatic speech recognition; convolutional neural network; connectionist temporal classification; End-to-End; neural network; erosion; recurrent neural network

Introduction

Speech recognition, often known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, is a capability enabling software to transform spoken language into written text. Although it is occasionally mistaken for voice recognition, speech recognition specifically concentrates on converting speech from a spoken to a textual format, distinguishing it from voice recognition, which aims solely to identify the voice of a particular individual.

Different speech technology applications are being used by a wide range of industries today, which helps both businesses and consumers save time and even lives.

Various methods are used to create automatic speech recognition. These methods include Data Time Wrapping (DTW), Hidden Markov Model (HMM), Dynamic Bayesian Network (DBN), ANN and deep neural network (DNN) (Hebash H.O. Nasereddin, January 2018). HMM and NN are the most widely used methods in recent times for speech recognition. HMM works by breaking down speech signals into a sequence of states and then using the acoustic properties of the speech to determine the likelihood of a sequence of phones (Kebebew, 2010).

Although neural networks (Bourlard & Morgan, 1993; Hinton et al., 2012) have significantly assisted automatic speech recognition, they now make up only one portion of a complicated pipeline. Like in conventional computer vision, the first step of the pipeline is the extraction of input features. Common methods include vocal tract length normalization and Mel-scale filterbanks (Davis & Mermelstein, 1980) (with or without a further transform into Cepstral coefficients) (Lee & Rose, 1998).

 doi:10.20944/preprints202402.0813.v1

Then, emission probabilities for a hidden Markov model are reconstituted from the neural network output distributions. The neural networks are then trained to recognize specific audio input frames (HMM).

Consequently, the actual performance metric and the goal function used to train the networks are very different (sequence-level transcription accuracy). This is the kind of consistency that end-to-end learning is designed to avoid. The fact that a considerable improvement in frame accuracy can lead to a little improvement or even a deterioration in transcription accuracy puzzles researchers. Another problem is that the frame-level training targets must be inferred from the alignments the HMM obtained. As a result, in an uncomfortable iterative process, network retraining and HMM realignments are alternated to provide targets that are more exact. Direct training of HMM neural network hybrids has been done using full-sequence training techniques such as maximum mutual information to increase the likelihood of accurate transcription (Bahl et al., 1986; Jaitly et al., 2012). However, these methods can only be used to retrain a system that has already been trained at the frame level, and they necessitate the careful adjustment of several hyper-parameters, often much more than for deep neural networks. The goals provided to the networks are often phonetic, despite the fact that the transcriptions used to train speech recognition systems are lexical. To translate words into phoneme sequences, a pronunciation dictionary is required. Such dictionaries require a lot of human effort to create, and often have a great impact on performance (Graves, A., & Jaitly, N., 2014). Another source of expert information, "state tying," is required to lessen the number of target classes since coarticulation effects are taken into account by multiphone contextual models, which adds another layer of complexity.

Existing Amharic speech recognition systems are composed of several different components, including a feature extraction module, an acoustic model, a language model, and a decoder. These systems require significant amounts of data and manual feature engineering, which can be time-consuming and labor-intensive. On the contrary, end-to-end speech recognition systems use a single neural network to map an input speech signal directly to an output transcript. These systems require less manual feature engineering and can be trained on raw speech signals, making them more efficient and effective.

Speech recognition has been a challenging problem in the field of artificial intelligence for decades, and traditional systems rely on complex pipelines of feature extraction, acoustic modeling, and language decoding. However, recent advances in deep learning have allowed for the development of end-to-end speech recognition models, that can directly transcribe speech to text without the need for intermediate steps.

The system proposed in this research replaces as much of the speech pipeline with a single recurrent neural network (RNN) architecture. While it is possible to directly transcribe unprocessed speech waveforms using RNNs (Graves, 2012, Chapter 9) or features learned using limited Boltzmann machines (Jaitly & Hinton, 2011), the computational cost is significant, and the performance typically lags behind conventional preparation. As a result, we have decided to use spectrograms as the minimum required preprocessing technique.

This research addresses important issues in addition to solving scientific issues. The difficulties that hearing-impaired people have in understanding other people's speeches makes it difficult for them to interact with non-hearing-impaired people, which prevents them from learning about their surroundings. Our work speeds up by typing directly from human voice, which is great for people who struggle with precise word placement. We therefore came up with the notion of creating an end-to-end speech recognition model that transforms speech to text to get around those difficulties and make life easier.

To the best of our knowledge, previous speech recognition models for Amharic language are built using traditional speech recognition mechanisms using acoustic, pronunciation, language models with a relatively smaller number of data, without considering a single pipeline and automatic feature extraction (Baye, A., Tachbelie, Y. & Besacier, L.,2021). Even currently available end-to-end trials in the Amharic language are applied using language and acoustic models separately, and their feature extraction methods do not utilize neural networks. But we propose an end-to-end speech

 doi:10.20944/preprints202402.0813.v1

recognition mechanism, which enables us to directly convert speech to text by replacing those traditional pipelines with a single RNN pipeline. Therefore, unlike traditional HMM based speech recognition models, our speech recognition model will not have those individual pipelines, for example, pronunciation dictionaries are not needed in our case. So, our study saves the time spent for preparing those dictionaries and finding domain expertise on certain areas. End-to-end is a system which directly maps a sequence of input acoustic features into a sequence of graphemes or words. We are expecting that our end-to-end speech recognition model will greatly simplify the complexity of traditional speech recognition. With the advances in neural networks, the need for manual labeling of language and pronunciation information is significantly reduced, as the neural network can now autonomously learn and capture such information. According to the literature, there are two main structures for end-to-end speech recognition (Baye, A., Tachbelie, Y. & Besacier, L. ,2021); attention model and CTC. We have used CTC in our case and it has solved the alignment problem that occurs in traditional models.

Materials & Methods

We got 110 hours from Andreas Nürnberger - Data and Knowledge Engineering Group and 20 hours from the ALFFA project. We have used 62 hours and 30 minutes, clipped from VOA and DW radios, from a previous project of our own work. We augment these noise-free read speech audios using time stretching, pitch shifting, speed perturbation, time and pitch scaling, dynamic range compression, filtering, time shifting, and amplitude scaling to make the whole audio tally 1732 hours and 30 minutes. The audio data obtained from Andreas Nürnberger - Data and Knowledge Engineering Group includes transcriptions written in English characters. To make the transcriptions compatible with Amharic, we performed a transcription process, converting them into Amharic characters. Additionally, we utilized the transcriptions created previously for our personal project. This process involved carefully listening to the audio and converting it into written text. The text data serve as the ground truth for the deep learning model, enabling it to learn the relationship between the audio features and the corresponding text. We used a frame length of 256 samples and a frame step of 160 samples to extract audio features.

The data is converted to spectrograms using Short-Time Fourier Transform (STFT) for feature extraction. The resulting spectrograms are then used as input to the neural network. Overall, the research aims to contribute to the development of improved speech recognition and processing technologies.

Training and Validation Phases: The training phase is a critical step in developing our end-to-end speech recognition model. It involves training a neural network model, specifically a combination of recurrent neural network (RNN) variants and neural network (CNN), to recognize speech patterns and convert them into text. The training data, which consists of a mixture of noisy-free and noisy speech data, are used to adjust the model's parameters and improve its accuracy. The training process continues until the model achieves satisfactory accuracy in the training data.

Following the training phase, the validation phase is conducted to evaluate the system performance on unseen data. For this purpose, a separate validation data set, derived from the training data, is used. The goal of the validation phase is to monitor the system's performance and prevent overfitting, which refers to a situation where the model performs well on the training data but poorly on new, unseen data. The accuracy of the predicted transcription on the validation data are measured using the Word Error Rate (WER) metric. The model's hyperparameters, such as the learning rate, number of layers, and number of neurons, are adjusted during the validation phase to improve the accuracy of the validation data.

Building the Deep Learning Model: To build the Amharic speech recognition model, a deep learning algorithm is applied to the collected and processed data. The model is based on a hybrid approach that combines a CNN with an RNN and utilizes a Connectionist Temporal Classification (CTC) loss function.

The model architecture consists of several key components. The input to the model is a representation of the audio data. The input is passed through a series of convolutional layers, which

 doi:10.20944/preprints202402.0813.v1

apply filters to extract relevant features from the spectrogram. Batch normalization and ReLU activation functions are used after the layers to enhance network performance.

After the layers, the output is fed into bidirectional GRU layers. These layers capture temporal dependencies by processing the sequence in both forward and backward directions. The outputs of the GRU layers are concatenated and passed through a fully connected layer. The model's training is guided by the CTC loss function, which aligns the predictions with the target labels without requiring the input data.

Model Implementation: The implementation of the model follows a TensorFlow/ framework. The input to the model is a variable-length sequence of spectrograms, reshaped to include an additional channel dimension for 2D layers. The customized CNN architecture includes two layers that extract useful features from the input audio spectrogram.

The reshaped output of the second layer is prepared for the recurrent layers by collapsing the height and width dimensions into a single dimension. This ensures proper processing of the features in the recurrent layers.

The recurrent layers consist of bidirectional Gated Recurrent Units (BiGRUs) with tanh activation functions and sigmoid recurrent activations. The number of units in each GRU is specified by the argument. Dropout is applied after each bidirectional layer, except for the last one.

The combination of convolutional layers, batch normalization, and ReLU activation functions in the CNN architecture helps the model to learn useful local features from the input spectrogram, which are then used by the recurrent layers to generate transcriptions of the input speech signal.

Overall, the model undergoes training and validation phases, with adjustments made to hyperparameters during the validation phase to improve accuracy. The implementation of the model involves the use of convolutional and recurrent layers to process the input spectrograms, along with appropriate reshaping and activation functions to facilitate feature extraction and modeling of temporal dependencies.

The architecture of the proposed model is presented in Figure 1 below.

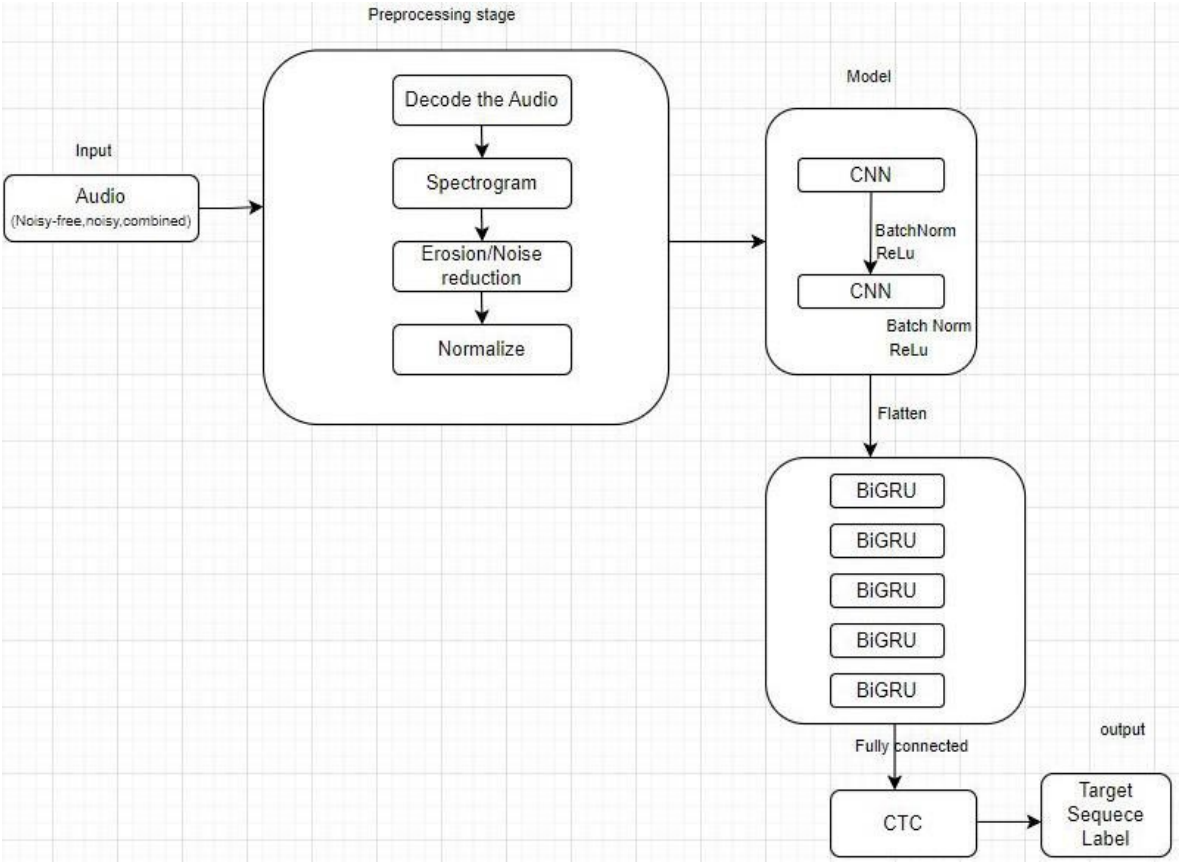

Figure 1. Proposed architecture of the model.

 doi:10.20944/preprints202402.0813.v1

Evaluation Metric

WER is a widely used metric to evaluate speech recognition systems. It measures the percentage of incorrectly recognized words compared to the reference transcription. Substitution, deletion, and insertion errors are considered, and WER is calculated by adding these errors and dividing by the total number of words in the reference. Lower WER indicates better performance and allows for model comparison and hyperparameter tuning.

Results and Discussion

We evaluated the performance of the speech recognition model using the word error rate (WER), calculated as the percentage of words incorrectly recognized by the system. The results of our experiments are presented and the WER for the huge dataset was 2%. This's presented in Figure 2 below.

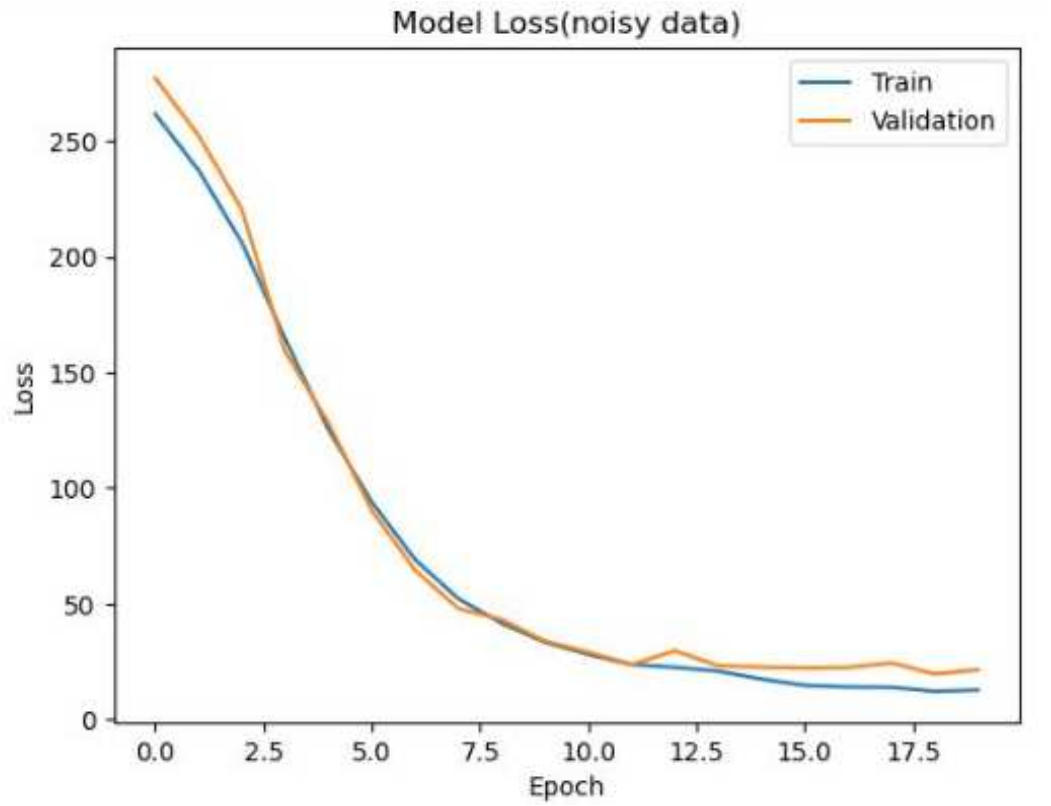

Figure 2. presents the loss function of our model.

In the context of our end-to-end speech recognition model, the x-axis of a plot in the figure typically represents the number of training epochs, which refers to the number of times the entire training dataset is fed to the model for learning. The y-axis, on the other hand, represents the loss, which is a measure of how well the model is performing in predicting the correct output for a given input. The loss function used in speech recognition tasks typically quantifies the difference between the predicted output and the actual output for a given input.

At the beginning of the training process, the loss value is usually high as the model has not yet learned to accurately predict the output of the input data. As the model is trained on more data and the number of epochs increases, the loss gradually decreases, indicating that the model is becoming better at predicting the output. This decrease in loss can be attributed to the model learning patterns and features in the training data, which allows it to make better predictions.

In the plotted graph, we can observe that the loss value gradually stabilizes and converges to a low value as the training progresses. This indicates that the model has learned to accurately predict the output of the input data and has reached a state of convergence. The point at which the loss

 doi:10.20944/preprints202402.0813.v1

stabilizes and converges can vary depending on various factors, such as the size and complexity of the dataset, the architecture of the model, and the training hyperparameters.

Although training loss is a good measure of how well the model fits the training data, it is not always a good indicator of how well the model will perform on new and unseen data. Here the validation loss comes into play. The validation loss is calculated by evaluating the model's performance on a separate set of data that it has not seen during training. Typically, validation data is a subset of the entire dataset that is held out specifically for this purpose.

By comparing training and validation losses, we can gain valuable insight into the performance and generalizability of our speech recognition model. During training, if the model is overfitting to the training data, we might observe that the training loss continues to decrease while the validation loss starts to increase, indicating that the model is becoming less accurate in predicting the output for new data. However, if the model is underfitting to the training data, we might observe that both the training and validation losses are high, indicating that the model is not learning the patterns and features in the data effectively. As we analyze our model, it becomes apparent that the loss metric consistently stabilizes and eventually reaches a low value as training progresses. This pattern of convergence demonstrates that the model has acquired the ability to make precise predictions for the given input data, and it has attained a state of convergence.

Assessing the performance and generalization ability of our speech recognition model relies heavily on examining the correlation between its training and validation losses. The training loss denotes the level of fitness of the model to the training data, whereas the validation loss signifies how well the model is likely to perform on fresh, previously unseen data. By keeping track of the training and validation losses, we can make informed choices regarding the model architecture and training hyperparameters, which in turn can enhance the model's performance and generalization ability.

Adam optimizer is preferred for end-to-end speech recognition due to its effectiveness in handling large-scale datasets and complex models. It combines the benefits of both AdaGrad and RMSProp algorithms by adapting the learning rate for each parameter individually. This adaptive learning rate adjustment helps in efficient optimization and convergence, making it suitable for speech recognition tasks.

A learning rate of 0.0001 is chosen for the end-to-end speech recognition to strike a balance between learning speed and accuracy. A lower learning rate allows for finer adjustments to the model's parameters, which can help in achieving better convergence and avoiding overshooting the optimal solution. It helps stabilize the training process and prevent drastic updates that may lead to sub optimal performance.

In the context of this end-to-end speech recognition, a drop rate of 0.5 typically refers to the dropout regularization technique. Dropout randomly sets a fraction of input units to 0 during training, which helps prevent overfitting and improves the model's generalization ability. A dropout rate of 0.5 means that, on average, half of the input units are dropped during training, providing regularization to the network. This helps prevent the model from relying too heavily on specific input features, leading to a more robust and accurate speech recognition system.

Our work has demonstrated the effectiveness of our end-to-end speech recognition models in large amount of data, with the model achieving exceptional accuracy. These results can have important implications for a variety of applications, such as improving accessibility for individuals with hearing impairments or enhancing the accuracy of voice-controlled devices in controlled environments.

We first evaluated the performance of our speech recognition model without erosion. When we applied the erosion technique to enhance the spectrogram representation of the input audio, we achieved an impressive Word Error Rate (WER) of 1.9%. This indicates that our model excels at accurately recognizing speech in clean environments. Even without erosion, the model performed remarkably well with a WER of 2.1%.

A WER displaying for this model and the predictions for unseen validation data via the RTX800 NVIDIA GPU is presented in Figure 3. below.

 doi:10.20944/preprints202402.0813.v1

Figure 3. Presents a WER displayed and the predictions for unseen validation data via RTX800 NVIDIA GPU.

Error analysis

We analyzed the errors made by the model on the test sets to identify common errors occurred.

The model performed well, with a WER of 2%. It performs exceptionally good in test data of the dataset. However, during our evaluation of the model's performance on a newly recorded audio from a natural environment, we noticed that it encountered difficulty in accurately predicting characters that possess similar visual representations. During the error analysis, it's observed that the model occasionally exhibited character swaps in its transcriptions. Specifically, certain characters were substituted with similar looking characters, leading to errors in the output. One common swap observed was the substitution of the character "ከ" (ke) with "ቀ" (q'a).

These characters have similar visual representations in their spectrogram representation.

As a result, the model sometimes mistakenly replaced instances of "ከ" with "ቀ" in its transcriptions, which could introduce inaccuracies. Similarly, another swap involved the characters "ተ" (te) and "ጠ" (t'e). These characters share similar visual features in their visual representation of their audio. Consequently, the model occasionally misinterpreted "ተ" as "ጠ" and vice versa, leading to incorrect transcriptions.

Another notable swap occurred between the characters "ቸ" (ch') and "ፏ" (tch'). These characters bear resemblance in terms of their visual structure in spectrogram and became a challenge to CNN. As a result, the model occasionally confused "ች" with "ፏ," resulting in errors in the transcribed text.

Conclusions

This study addresses the challenges of traditional automatic speech recognition (ASR) methods by proposing an approach that utilizes a single recurrent neural network (RNN) architecture. The objective was to streamline the speech recognition pipeline and improve the efficiency and accuracy of the system.

The conventional ASR pipeline often requires multiple separate components, such as language, acoustic, and pronunciation models with dictionaries, resulting in time- consuming processes and performance limitations. Using the power of RNNs, our proposed end-to-end system significantly simplifies this pipeline.

Our research findings indicate that applying erosion to the spectrograms has a positive effect on speech recognition and enhances model performance, although the improvement is not significant.

In building an end-to-end speech recognition model for Amharic, we selected BiGRU as the preferred deep learning algorithm. This decision was based on the observation that BiLSTM required approximately twice the processing time of BiGRU and involved a greater investment in computational resource.

Through rigorous evaluation using the word error rate (WER) metric, our approach demonstrated impressive performance. We achieved a remarkable WER of 2%, showcasing the system's robustness in clean environments.

This research has important implications for the field of speech recognition. By reducing the need for manual efforts in creating dictionaries and integrating multiple models, our approach not only saves time but also enhances the practicality of ASR systems for real- world applications. The efficiency and accuracy improvements brought forth by our end- to-end RNN-based architecture pave the way for more accessible and effective speech recognition solutions.

This research successfully achieved the main objective of developing an end-to-end speech recognition model for the Amharic language using deep learning. The architecture of the model combines a convolutional neural network (CNN) with a recurrent neural network (RNN) and utilizes a connectionist temporal classification (CTC) loss function.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were not required for this study as it did not involve human or animal subjects. This research was conducted as part of the thesis research at Bahir Dar Institute of Technology.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data we use for this study is available openly at fig share in the following link. https://figshare.com/articles/dataset/Yohannes_A_Ejigu_Amharic_ASR_Dataset_zip/24959727.

Conflicts of Interest: We declare that there are no conflicts of interest to disclose.

References

1. Abate, S. T. (2005). Automatic Speech Recognition for Ahmaric, PhD dissertation. German: Hamburg University.
2. Abebe Tsegaye(2019)**.** *Designing automatic speech recognition for ge'ez language*. Bahir Dar, Ethiopia: Master's thesis.
3. Azmeraw Dessalegn (2019). Syllable based speaker independent Continous speech recognition for Afan Oromo. Bahir Dar, Ethiopia: Master's thesis.
4. Bahl, L., Brown, P., De Souza, P.V., and Mercer, R.(1986). *Maximum mutual information estimation of hidden markov model parameters for speech recognition*. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86., volume 11, pp. 49–52. doi: 10.1109/ICASSP.1986.1169179.
5. Bisani (2005), Maximilian and Ney, Hermann. Open vocabulary speech recognition with flat hybrid models. In INTERSPEECH, pp. 725–728.
6. Bourlard, Herve A. and Morgan, Nelson (1993). *Connectionist Speech Recognition: A Hybrid Approach. Kluwer Academic Publishers*, Norwell, MA, USA, ISBN 0792393961.
7. Davis, S. and Mermelstein, P.(1980) *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. IEEE Transactions on Acoustics, Speech and Signal Processing, 28(4):357–366.
8. Baye, A., Tachbelie, Y., & Besacier, L. (2021). End-to-end Amharic speech recognition using LAS architecture. In Proceedings of the 12th Language Resources and Evaluation Conference (pp. 4058-4065). European Language Resources Association.
9. Dumitru, C. O., & Gavat, I. (2006, June). *A Comparative Study of Feature Extraction Methods Applied to Continuous Speech Recognition in Romanian Language*. In Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006 focused on (pp. 115-118). IEEE.
10. Furui, S., Ichiba, T., Shinozaki, T., Whittaker, E. W., & Iwano, K. (2005). *Cluster-based modeling for ubiquitous speech recognition. Interspeech* 2005, 2865-2868.
11. Galescu, Lucian (2003). Recognition of out-of-vocabulary words with sub-lexical language models. In INTERSPEECH.
12. Gebremedhin, Y. B., Duckhorn, F., Hoffmann, R., & Kraljevski, I. (2013). *A new approach to develop a syllable based , continuous amharic speech recognizer*, (July), 1684–1689.
13. Graves, Alex.(2012) *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of Studies in Computational Intelligence. Springer.

14. Hebash H.O. Nasereddin, A. A. (January 2018). Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation. 2017 computing conference. London, UK: IEEE.
15. Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George, rahman Mohamed, Abdel, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara, and Kingsbury, Brian (2012). *Deep neural networks for acoustic modeling in speech recognition.* Signal Processing Magazine.
16. Jaitly, Navdeep and Hinton, Geoffrey E.(2011) Learning a better representation of speech soundwaves using restricted boltzmann machines. In ICASSP, pp. 5884–5887.
17. Jitendra Singh Pokhariya, D. S. (2014). *Sanskrit Speech Recognition using Hidden Markov Model Toolkit* . International Journal of Engineering Research & Technology , 3(10).
18. Kebebew, T. (2010). speaker dependent speech recognition for afan oromo using hybrid hidden markov models and artificial neural network. Addis Ababa, Ethiopia: Addis Ababa University.
19. Solomon Teferra Abate, W. M. (n.d.). *Automatic Speech Recognition for an UnderResourced Language* – Amharic. Department of Informatics, Natural Language Systems Group, University of Hamburg, Germany.
20. Teferra, S., & Menzel, W. (2007). *Syllable based speech recognition for Amharic*, (June), 33–40
21. Tohye, T. G. (2015). *Towards Improving the Performance of Spontaneous Amharic Speech Recognition* , master thesis. Addis ababa, Ethiopia: Addis Ababa University.
22. Yifru, M. (2003). *Application of Amharic speech recognition system to command.* Addis Ababa, Ethiopia: Master's thesis, Addis.
23. A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In ICML, pages 369–376. ACM.
24. Jurafsky, D., & Martin, J. H. (2009). Speech and language processing (2nd ed.). Prentice Hall.
25. Huang, X., Acero, A., & Hon, H. W. (2001). Spoken language processing: A guide to theory, algorithm, and system development. Prentice Hall PTR.
26. Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 6645–6649).
27. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). *Deep neural networks for acoustic modeling in speech recognition:* The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82–97.
28. Graves, A., & Jaitly, N. (2014). *Towards end-to-end speech recognition with recurrent neural networks.* In International Conference on Machine Learning (pp. 1764–1772).
29. Jelinek, F., & Mercer, R. L. (1980). *Interpolated estimation of Markov source parameters from sparse data.* In Proceedings of the Workshop on Pattern Recognition in Practice (pp. 381–397)
30. Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing.
31. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., & Woodland, P. (2006). *The HTK Book (for HTK Version 3.4).* Cambridge University Engineering Department.
32. Abdel-Hamid, O., & Jiang, H. (2013). *Fast speaker adaptation of deep neural networks using model compression*. In Proceedings of the 2013 IEEE international conference on acoustics, speech and signal processing (pp. 7822-7826). IEEE.
33. Chavan, S., & Sable, P. (2013). Speech recognition using hidden markov model.
34. International Journal of Engineering Research and Applications, 3(4), 2319-2324.
35. Novoa, J., Lleida, E., & Hernando, J. (2018). A comparative study of HMM-based and DNN-based ASR systems for under-resourced languages. Computer Speech & Language, 47, 1-22.
36. Palaz, D., Kılıç, R., & Yılmaz, E. (2019). A comparative study of deep neural network and hidden Markov model based acoustic models for Turkish speech recognition. Journal of King Saud University-Computer and Information Sciences
37. Lee, K., & Kim, H. (2019). Dynamic Wrapping for Robust Speech Recognition in Adverse Environments. IEEE Access, 7, 129541-129550.
38. Wang, Y., & Wang, D. (2019). *A novel dynamic wrapping method for robust speech recognition under noisy environments.* Journal of Ambient Intelligence and Humanized Computing, 10(5), 1825-1834.
39. Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). *Attention- based models for speech recognition.* In Advances in neural information processing systems (pp. 577-585).

40. Kim, J., Park, S., Kang, K., & Lee, H. (2017). Joint CTC-attention based end-to-end speech recognition using multi-task learning.
41. Zhang, Y., Deng, K., Cao, S., & Ma, L. (2021). Improving Hybrid CTC/Attention End-to- end Speech Recognition with Pretrained Acoustic and Language Model.
42. Deng, K., Zhang, Y., Cao, S., & Ma, L. (2022). Improving CTC-based speech recognition via knowledge transferring from pre-trained language models.
43. Chen, C. (2021). ASR Inference with CTC Decoder. PyTorch.
44. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., … & Prenger, R. (2014). *Deep speech: Scaling up end-to-end speech recognition*
45. Igor Macedo Quintanilha (2017). End-to-end speech recognition applied to brazilian portuguese using deep learning
46. Hanan Aldarmaki, Asad Ullah, and Nazar Zaki (2021). *Unsupervised Automatic Speech Recognition: A Review.* arXiv.org, 2021, arxiv.org/abs/2106.04897
47. Abdelrahma Ahmed, Yasser Hifny,Khaled Shaalan and Sergio Tolan(2016). *Lexicon Free Arabic Speech Recognition*. https://link.springer.com/chapter/10.1007/978-3-319-48308- 5_15
48. Wren, Y., Titterington, J., & White, P. (2021). How many words make a sample? Determining the minimum number of word tokens needed in connected speech samples for child speech assessment. Clinical Linguistics & Phonetics, 35(8), 761-778. https://doi.org/10.1080/02699206.2020.1827458
49. Malik, M., Malik, M. K., Mehmood, K., & Makhdoom, I. (2021). *Automatic speech recognition: a survey. Multimedia Tools and Applications,* 80(6), 9411-9457. https://doi.org/10.1007/s11042-020-10073-7
50. Zhang, X., & Wang, H. (2011). *A syllable-based connected speech recognition system for Mandarin*. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4856-4859).
51. IEEE. https://doi.org/10.1109/ICASSP.2011.5947265
52. Khan, A. I., & Zahid, S. (2012). *A comparative study of speech recognition techniques for Urdu language*. International Journal of Speech Technology, 15(4), 497-504. http://doi.org/10.1007/s10772-012-9175-z
53. Chen, C.-H., & Chen, C.-W. (2010). *A novel speech recognition system based on a self- organizing feature map and a fuzzy expert system*. Microprocessors and Microsystems, 34(6), 242-251. http://doi.org/10.1016/j.micpro.2010.04.003
54. Ge, X., Wu, L., Xia, D., & Zhang, P. (2013). A multi-objective HSA of nonlinnear system modeling for hot skip-passing.
55. In 2013 Sixth International Conference on Advanced Computational Intelligence (ICACI) (pp. 1-6). IEEE. https://doi.org/10.1109/ICACI.2013.6748484
56. Villa, A. E. P., Masulli, P., & Pons Rivero, A. J. (Eds.). (2016). *Artificial neural networks and machine learning* – ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part I. Springer International Publishing. https://doi.org/10.1007/978-3-319-44781-0
57. Chen, X., & Hu, Y. (2012*). Learning optimal warping window size of DTW for time series classification*. In 2012 11th International Conference on Information Science,
58. and their Applications (ISSPA) (pp. 1-4). IEEE. https://doi.org/10.1109/ISSPA.2012.6310488
59. La Rosa, M., Loos, P., & Pastor, O. (Eds.). (2016). Business Process Management: 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings. Springer. https://doi.org/10.1007/978-3-319-45348-4
60. Nguyen, V. N. (2016). *A framework for business process improvement: A case study of a Norwegian manufacturing company* (Master's thesis). University of Stavanger. https://brage.bibsys.no/xmlui/bitstream/id/430871/16-00685- 3%20Van%20Nhan%20Nguyen%20- %20Master's%20Thesis.pdf%20266157_1_1.pdf
61. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-based learning applied to document recognition*.
62. *Proceedings of the IEEE*, *86*(11), 2278-2324. https://doi.org/10.1109/5.880079
63. Ientilucci, E. J. (2006). Statistical models for physically derived target sub-spaces. In S. S. Shen & P. E. Lewis (Eds.), *Imaging Spectrometry XI* (Vol. 6302, p. 63020A). SPIE. https://doi.org/10.1117/12.679525
64. Begel, A., & Bosch, J. (2013). The DevOps phenomenon. *Communications of the ACM*, *56*(11), 44-49. http://dl.acm.org/citation.cfm?id=2113113.2113141
65. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press

 doi:10.20944/preprints202402.0813.v1

66. Scherzer, O. (Ed.). (2015). *Handbook of mathematical methods in imaging* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-46478-7
67. Persagen Consulting. (n.d.). *Machine learning*. https://persagen.com/files/ml.html
68. Scherzer, O. (Ed.). (2015). *Handbook of mathematical methods in imaging* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-46478-7
69. Saha, S. K., & Tripathy, A. K. (2017). Automated IT system failure prediction: A deep learning approach. *International Journal of Computer Applications, 159*(9), 1–6. https://www.researchgate.net/publication/313456329_Automated_IT_system_failure_prediction_A_deep_learning_approach
70. Kotera, J., Šroubek, F., & Milanfar, P. (2013). *Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors.* In A. Petrosino, L. Maddalena, & P. Soda (Eds.), Computer analysis of images and patterns (pp. 59–66). Springer. https://doi.org/10.1007/978-3-642-40246-3_8
71. Artificial intelligence. (2022, January 5). In Wikipedia.
72. https://en.wikipedia.org/wiki/Artificial_intelligence
73. Mwiti, D. (2019, September 4). A 2019 guide for automatic speech recognition. Heartbeat.
74. https://heartbeat.fritz.ai/a-2019-guide-for-automatic-speech-recognition-f1e1129a141c
75. Persagen. (n.d.). *Machine learning*. *Retrieved* January 6, 2022, from https://persagen.com/files/ml.html
76. Klein, A., & Kienle, A. (2012). *Supporting distributed software development by modes of collaboration.* Multimedia Systems, 18(6), 509–520. http://doi.org/10.1007/s00530-012-0266-0
77. Chen, Y., & Cong, J. (2017). DAPlace: Data-aware three-dimensional placement for large-scale heterogeneous FPGAs.
78. In 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 236–243). IEEE. http://doi.org/10.1109/ICCAD.2017.8203812
79. Zhang, Y., Zhang, H., Lin, H., & Zhang, Y. (2017). *A new method for remote sensing image registration based on SURF and GMS.* Remote Sensing, 9(3), 298. http://doi.org/10.3390/rs9030298
80. Schmidhuber, J. (2000). *How to count time: A simple and general neural mechanism*. ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf
81. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014).
82. *Generative adversarial nets*. arXiv. https://arxiv.org/pdf/1406.1078v3.pdf
83. Srivastava, A. (2018, November 14). *Basic architecture of RNN and LSTM. PyDeepLearning.* https://pydeeplearning.weebly.com/blog/basic-architecture-of-rnn-and- lstm
84. Graves, A., Mohamed, A.-R., & Hinton, G. (2013*). Speech recognition with deep recurrent neural networks.* 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6645–6649. https://www.cs.toronto.edu/~graves/asru_2013.pdf
85. Zhang, Y., Zhang, J., Zhang, J., & Li, H. (2020). *A survey on deep learning for named entity recognition*. IEEE Transactions on Knowledge and Data Engineering, 32(9), 1635– 1658. https://doi.org/10.1109/TKDE.2019.2906425
86. Papers with Code. (n.d.). BiLSTM. https://paperswithcode.com/method/bilstm Olah, C. (2015, August 27).
87. Understanding LSTM networks. http://colah.github.io/posts/2015-08-Understanding- LSTMs/