International Journal on Engineering Technology (InJET)

www.kec.edu.np/journal
Volume 2, issue no. 1, Dec 2024, Page 96-105
Received Date: July 19, 2024
Accepted Date: Sept 23, 2024
ISSN: 3021-940X (Print)
DOI: https://doi.org/10.3126/injet.v2i1.72525

Advancements in Nepali Speech Recognition: A Comparative Study of BiLSTM, Transformer, and Hybrid Models

Ankit Kafle[1,*], Jenith Rajlawat[2], Nawaraj Shah[3], Neetish Paudel[4], Bishal Thapa[5]

[1] *Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, kafleankit798@gmail.com*
[2] *Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, jenithrajlawat@gmail.com*
[3] *Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, shah.nawaraj.ns@gmail.com*
[4] *Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, paudelneetish@gmail.com*
[5] *Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, bishalthapa@kec.edu.np*

Abstract

In today's world, leveraging Automatic Speech Recognition (ASR) technology to process and understand spoken language is highly desirable. Our proposed Nepali Speech Recognition employs an advanced generation to recognize and interpret spoken Nepali language. It approaches Nepali speech, allowing it to reply to user queries effectively. To attain this, we rent a mixture of superior neural network fashions. We extract Mel-frequency cepstral coefficients (MFCCs) from the preprocessed audio information; these MFCCs capture crucial spectral characteristics of Nepali speech and serve as essential input features for our neural network model. To design a top-rated version for textual content-based query processing, we make use of convolutional neural networks (CNN), residual networks (ResNet), and bidirectional long short-term memory (BiLSTM) layers. The CNN layers excel at extracting neighborhood patterns and spatial features from the MFCC input; the ResNet layers capture deeper representations to enhance performance. The BiLSTM layers are also employed to model temporal dependencies in the textual content-based query processing, we make use of convolutional neural networks (CNN), residual networks (ResNet), and bidirectional long short-term memory (BiLSTM) layers. The CNN layers excel at extracting neighborhood patterns and spatial features from the MFCC input; the ResNet layers capture deeper representations to enhance performance. The BiLSTM layers are also employed to model temporal dependencies in the textual content records. We hired the Connectionist Temporal classification (CTC) loss feature to enable sequence-to-series mapping, aligning the input speech with corresponding text outputs. This approach permits our gadget to successfully process textual content queries and provide correct responses, enhancing the user's usefulness. The model, after being trained with 1.55 million parameters in about 1 lakh 57 thousand audio datasets for 47 epochs, achieved a CTC of 17.98% (82.02%-character accuracy rate) with this model.

Keywords: Automatic Speech Recognition, Convolutional Neural Networks, Connectionist Temporal Classification, Mel-frequency cepstral coefficients, Residual Networks, Bidirectional Long Short-Term Memory.

1. Introduction

Speaking and writing are crucial methods of communication. Deficiencies in either can impact daily life. Many individuals in rural areas can speak well but struggle with writing. Communication technologies often require text input making familiarity with tools like Automatic Speech Recognition (ASR) valuable (Bhatta, et al., 2020). ASR technology converts spoken language into written textual content and has advanced use in telecommunications, transcription services, virtual assistants, and voice-controlled systems. It changed

**Corresponding Author*

human-laptop interaction and allowed for more effective communication. ASR is necessary for numerous essential reasons. First, it permits a natural and intuitive method of interaction among humans and machines. Through speaking commands or questions in preference to typing or pressing buttons, users can interact with the system more easily, imparting convenience and accessibility. ASR also plays a key role in making digital content and offerings available to people with listening or speech impairments, enabling them to participate in conversations and get entry to data through captioning and transcription services.

Similarly, ASR systems that aid multiple languages guide powerful communication within language boundaries and promote global connectivity and expertise. ASR appreciably increases performance and productivity in various areas, automating obligations such as transcription and simplifying workflows. Looking ahead, the long-term impact of developing ASR systems, particularly in underserved regions, could be transformative. These technologies can play a key role in the digital inclusion of marginalized communities, allowing people with limited literacy or access to digital tools to engage more fully with information and services. Moreover, advancements in ASR could lead to broader technological breakthroughs, particularly in fields like natural language processing, machine learning, and artificial intelligence. In the long run, investing in such systems can foster greater social and economic participation, particularly in rural or underrepresented areas, ultimately contributing to the closing of the digital divide. This system seamlessly converts spoken queries to textual content, extracts and interprets user input, and facilitates information evaluation and insights by using processing and analyzing huge speech datasets, enabling precious record extraction and knowledge discovery. In addition, ASR automates transcription and simplifies creation. A part of ASR is Nepali Speech Recognition, where the Nepali speech provided by the user is converted into Nepali text and can be processed further for useful interpretation.

2. Problem Statement

Limited English proficiency among some Nepali-speaking populations hampers their ability to perform tasks requiring English communication. According to the latest census data 2021, 44.86% of the population recorded Nepali as their mother tongue. This language barrier not only limits access to essential services but also has broader social and economic implications, such as reduced employment opportunities, limited access to education, and social exclusion. Furthermore, while there are existing language translation tools, they often lack accuracy for Nepali speech because of the insufficient data of different speakers for different languages spoken, which makes it difficult to translate the speech into its proper format. These challenges highlight the urgent need for a user-friendly Nepali speech recognition system that accurately converts Nepali speech into text, bridging the gap and enabling broader access to opportunities and resources for those with limited English proficiency.

3. Related Works

One notable ASR software for Nepali is the system developed by the Center for Speech and Language Technology (CSLT) at the University of Colorado Boulder (Tibet Himalaya Initiative, 2006). This system converts Nepali speech into written text and has been trained on extensive Nepali speech data to enhance accuracy.

The Kaldi toolkit, an open-source ASR framework developed by Johns Hopkins University researchers in 2011, is a key contribution to the field (Kaldi ASR, 2011). It offers a comprehensive set of tools and libraries, making it a flexible platform for building advanced ASR systems. The toolkit's effectiveness provides a solid foundation for developing an ASR model for the Nepali language. The Kaldi toolkit, an open-source ASR framework developed by Johns Hopkins University researchers in 2011, is a key contribution to the field (Kaldi ASR, 2011). It offers a comprehensive set of tools and libraries, making it a flexible platform for building advanced ASR systems. The toolkit's effectiveness provides a solid foundation for developing an ASR model for the Nepali language.

The proposed model for Nepali speech recognition combines CNN, GRU and CTC networks with MFCC feature extraction, showing promising potential for accurately transcribing spoken Nepali speech into written

text. This technology has the capacity to enhance interaction with communication devices and improve communication across various fields. Although the model achieves an 11% Word Error Rate (WER) using a dataset from Open Speech and Language Resources, further research and refinement are needed to enhance accuracy and real-world applicability. The 1D-CNN component captures high-level features from MFCC representations, while the GRU component, a type of recurrent neural network, excels in modeling sequential data, enabling the model to learn temporal dependencies and enhance recognition accuracy by capturing the context and structure of the Nepali language.(Bhatta, et al., 2020).

4. Methodology

4.1. Working Mechanism

(Fig. 1) represents a block diagram of a Nepali Speech Recognition training model. The process begins with a training dataset (OpenSLR), which is subjected to preprocessing to clean and normalize the data. Following this, MFCCs (Mel-frequency cepstral coefficients) are extracted as features that represent the audio's frequency characteristics. These features are passed to the acoustic model (CNN), which captures patterns in the sound signals. The output is processed by CTC (Connectionist Temporal Classification) for decoding, aligning predictions with input sequences. Concurrently, a BiLSTM (Bidirectional Long Short-Term Memory) model works as a language model, handling temporal dependencies. The network weights are updated during training based on testing data, and the final accuracy is calculated.

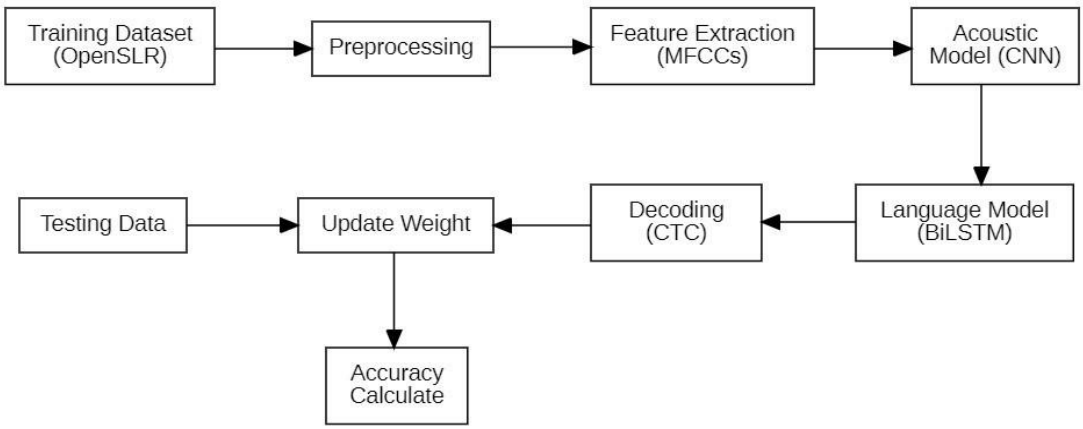

Figure 1. Block diagram of the Nepali speech recognition training model.

4.1.1 Data Acquisition and Cleaning

Two datasets were used: High-quality TTS data for Nepali (openslr.org/43) (Sodimana, et al., 2018) and a large Nepali ASR training dataset (openslr.org/54) (Kjartansson, et al., 2018). The first dataset contains female-transcribed audio data for the Nepali language. It includes 2064 high-quality audio files from 18 different speakers, with multiple audio files per speaker having similar word and character patterns. However, the vocabulary and text sequences might not be diverse, leading to bias in the ASR model's predictions if used. On the other hand, the second data set provides a significant advantage with a larger vocabulary, more speakers, a greater number of audio files, and a more diverse sequence of characters and text. This dataset was used for training and testing the ASR model. It consists of 157,905 audio clips from 527 unique speakers, sampled at 16 kHz. The dataset was sourced from OpenSLR, a platform hosting speech and language resources for speech recognition. Before training, an initial cleansing process was performed on the dataset by eliminating numeric transcriptions to prevent degradation of the model's overall performance. After removing these instances, approximately 143.6 hours of 148,188 audio clips remain as the foundation for training and testing the model. The dataset was acquired using OpenSLR, which provides a collection of speech and language resources. Firstly, the dataset underwent a cleansing process to eliminate numeric transcripts, as this type of data plays a minimal role and degrades the overall performance of the model. As

most of the dataset contained large silent gaps, the silent gaps were clipped. For cleaning, the window size was taken as 500. This process made the data more suitable for feature extraction and further processing, reducing errors.

ALGORITHM 1: Clipping of silent gap from both ends

```
wav ← sampled audio signal
Δ ← appropriate window length
INPUT: wav, Δ
PROCESS:
   wavAvg ← Average(|wav|)
   N ← Length(wav)
      /* Removing the silent gap from the start */
   for idx = 0, Δ, 2Δ, ..., N - Δ do
      win ← wav[idx : idx + Δ]
      winAvg ← Average(|win|)
      if winAvg > wavAvg then
         wav ← wav[idx :]
         break
      end if
   end for
      /* Removing the silent gap from the end */
   for idx = N - Δ, N - 2Δ, ..., 0 do
      win ← wav[idx : idx + Δ]
      winAvg ← Average(|win|)
      if winAvg > wavAvg then
         wav ← wav[: idx]
         break
      end if
   end for

 OUTPUT: processed_wav ← wav
```

4.1.2 Feature Extraction

After the Data cleaning part, the extraction of the best parametric representation of acoustic signals was an important task to produce better recognition performance. Mel Frequency Cepstral Coefficients (MFCCs) are used as a powerful feature extraction mechanism (Muda, et al., 2010). This feature extraction mechanism includes six stages. Pre-emphasis emphasizes higher frequencies, increasing the energy of the signal; framing segments of a speech signal into small duration blocks; and windowing reduces the discontinuity and smooths out the edges using a Hanning window function. The resultant is passed through a discrete Fourier transform to represent the frequency domain, and the Mel Filter Bank has a collection of bandpass filters over the Mel scale (i.e., 13). Mel scale measures the frequency of non-linear perception of pitch by human ears. The signal is passed through it, and the MFCCs finally obtained are converted to the time domain using a discrete cosine transform. (Fig. 2) displays a Mel-frequency cepstral coefficients (MFCCs) plot, which visualizes the frequency characteristics of an audio signal over time. The x-axis represents time (in seconds), while the y-axis shows the different MFCC coefficients. The x-axis represents time (in seconds), while the y-axis shows the different MFCC coefficients. The color scale on the right indicates the intensity, with warmer colors (red) representing higher energy levels or amplitude and cooler colors (blue) indicating lower energy. This visualization helps in identifying how different frequency components evolve, making it useful for speech and audio recognition tasks. The obtained results are quite favorable with multiple machine-learning components for signal processing. The equation involved in the calculation of the Mel scale from the frequency in Hertz (f) is given by:

$$Mel(f) = 2595 \cdot \log\left(1 + \frac{f}{700}\right) \quad (1)$$

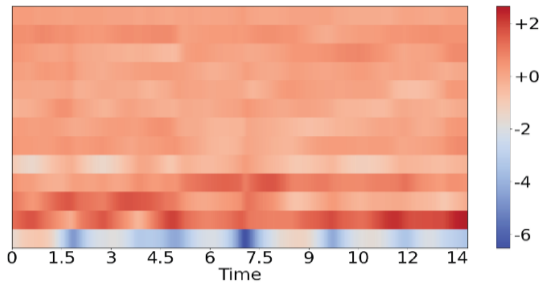

Figure 2. MFCCs of the audio where warmer colors (red) representing higher energy levels and cooler colors (blue) indicating lower energy.

4.1.3 Acoustic and Language Model

4.1.3.1 1D-CNN

After the residual block, the model employs a 1D-CNN layer for localized feature extraction. The 1D-CNN performed convolution operations on the temporal direction of the signal and used a trained weight filter (Kernels) to extract localized features. For the input in the 1D-CNN layer, the output shape was: (batch size: 2, sequence length: 1000, output dimension: 52).

4.1.3.2 ResNet

The ASR model begins with the implementation of residual blocks. The blocks utilized the shortcut connections to add the input of a block into the output of our stacked layers, such that the output will not be too skewed from the input. For the implementation of residual blocks in our model, there exists 1D-CNN and batch normalization (BN). 1D-CNN localized the features. BN adds stability and speed to gradient descent, speeds up training, and also normalizes the CNN layer's output vector. The activation function for the output of BN is parametric ReLU (PReLU). Lastly, the output of PReLU is added to the input of the residual block. The PReLU also doesn't change shape or dimensions, so its output was (2,100,50). (Fig. 3) illustrates a residual learning block commonly used in deep learning networks. The input x passes through three layers: a 1D Convolutional Neural Network (1D-CNN) layer that extracts features from the input data, followed by batch normalization, which normalizes these features to stabilize and speed up training. Afterward, a PReLU (Parametric Rectified Linear Unit) activation function is applied, which introduces non-linearity and helps the network learn more complex patterns. Simultaneously, the input x bypasses these layers through a shortcut connection and is added elementwise to the transformed output G(x), forming G(x)+xG(x). This skip connection allows the network to retain the original input information, making it easier to learn identity mappings and improving the training of very deep networks.

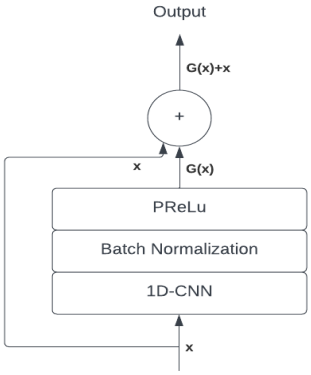

Figure 3. Residual Block: Input passes through 1D-CNN; output adds to input, enabling improved gradient flow and faster convergence during training.

4.1.3.3 RNN Layers

The last residual block's output was used as the source for multilayer RNN layers, primarily BiLSTM. An RNN is a type of neural network where the input for the current time step is the output from the previous time step. RNNs were able to capture contextual information throughout time and were appropriate for sequential data. In order to generalize patterns in data, RNN layers provide higher abstraction levels of features. Our model primarily uses two-way long-term short-term memory (BiLSTM), an RNN version. It concurrently processed the input sequence backward and forward, gathering data from both the past and the future. To capture complicated relationships, several BiLSTM layers were built on top of one another. As a result, the ASR model was better able to comprehend and accurately transcribe voice input. The output shape was (2,100,170) for both BiLSTM layers.

4.1.3.4 Dense Layers and SoftMax Output

The output of RNN layers (i.e., BiLSTM layers) was fed into the dense layers of the neural network, which further processed the features learned by RNNs. The output of the 1st Dense Layer output shape was (2,100,340), and after activation through ReLU and passing to the 2nd Dense Layer's output was (2,100,66), as the dimension of output 66 helps the softmax layer for probability calculation. Finally, a softmax layer was applied to obtain a probability distribution over the 66 unique characters. The softmax output can be represented as Softmax_output = Softmax(Dense (RNN_output)).

4.1.3.5 CTC (Connectionist Temporal Classification) Loss

The CTC loss function was used to compare the softmax output with the targeted transcriptions. The CTC loss helped handle the alignment problem between input audio and output characters. It computed an alignment-free loss value using a blank token introduced during training and inference. The CTC loss can be calculated as CTC_loss = -log (p (Y | X)), where p (Y | X) will represent the probability of the target transcription Y given the input audio X. And the objective function (i.e., probability p(Y | X)) is the sum of all possible valid sequences. Mathematically,

$$p(Y|X) = \sum_{A \in A_{x,y}} \left(\prod_{t=1}^{T} p_t(a_t|X) \right) \tag{2}$$

Where $A_{x,y}$ is the valid alignment of Y given X.

4.1.4 Decoding Algorithm

The SoftMax outputs were decoded to produce a character sequence during prediction. In our model, the CTC beam search decoding technique was applied. CTC Beam Search identified the most likely output sequence by taking into account many alignments at each time step. To choose the final sequence, a series of steps including tokenization, beam initialization, expansion, scoring, and pruning were performed. Accurate and contextually coherent transcriptions were produced by successfully addressing the problem of matching audio inputs with output characters using CTC Beam Search. (Fig. 4) represents the architecture of a Nepali Speech Recognition Model. In order to identify significant sound properties from short portions of the speech input, the Nepali Speech Recognition Model initially employs a 1D Convolutional Neural Network (1D-CNN). The result is then scaled for faster and more reliable learning by batch normalization. Repetitive residual blocks, which help the model learn efficiently by holding onto crucial information from previous layers, are applied five times to the input. Next, in order to extract context from the complete speech sequence, Bidirectional LSTMs (BiLSTM) process the sequence both forward and backward. ReLU adds non-linearity to the output after it has been refined by a thick layer, allowing the model to recognize more intricate patterns. In order to determine the likely words or phonemes in the detected speech, a Softmax output is used to translate the result into probabilities.

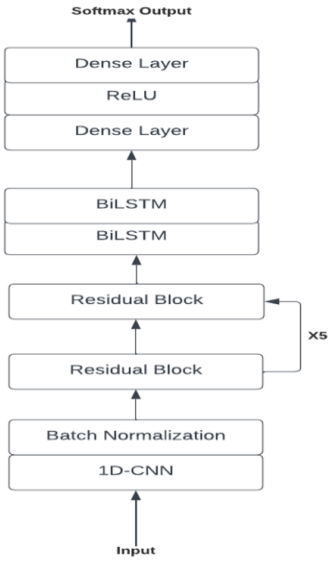

Figure 4. Model Architecture: Input passes through 1D-CNN and BiLSTM layers, producing a SoftMax output for classification

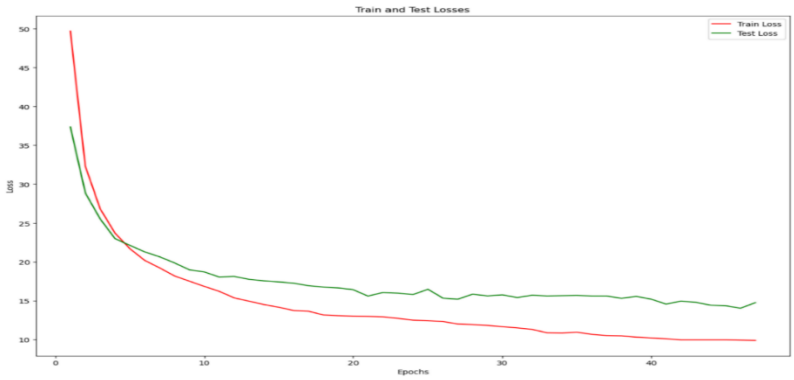

Figure 5. Train and test loss of trained model (1D-CNN+BiLSTM+RESNET)

Table 1. Models and their CER on test data

Model	Test Data CER	# Prams
BiLSTM	23.69%	1.12M
1D-CNN + ResNet + BiLSTM	17.06%	1.55M
Transformer CNN	22.72%	4.13M

Table 2. Models and their transcription of real time speeches

Actual Transcription	Model	Predicted Transcription
मानिसहरू अन्नाको समर्थनमा	1D -CNN+BiLSTM+RESNET	मानिसहरू अन्नाको सघर्थनमा
	Transformer CNN	मानिस र अन्नाको समर्थनमा
	BilLSTM	मानिस र अक्यगोसम्धनमा
गाविसहरूको लेखसँग यो	1D-CNN+BiLSTM+RESNET	गाविसहरूको लेखसँग यो
	Transformer CNN	गाविसहरूको लेखसँग यो

	BiLSTM	गाविसहरूको ले गयो
धर्के ढुकुर नेपालमा पाइने एक प्रकारको चराको नाम हो।	1D-CNN+BiLSTM+RESNET	ध्टे डुकुर नेपालमा पाइनी एक प्रकारको चराको नाम हो
	Transformer CNN	तट्कीय रूपकुर नेपालमा पाइन्य एक प्रकारको नेपालमा पाइन् हो।
	BiLSTM	तपेरुकु नेपालमा ऐइने एकव्यारको चलाकुदान हो
असोजमा भएको थियो	1D-CNN+BiLSTM+RESNET	असोजमा भएको थियो
	Transformer CNN	असमा भएको थियो
	BiLSTM	असरन भएको थियो
समयमा अघि चल	1D-CNN+BiLSTM+RESNET	समयमा अघि चल
	Transformer CNN	समयमा आयु चला
	BiLSTM	सरयाम आवि चल

5. Result and Discussion

We have completed the design and development of the project, successfully obtaining the desired output of transcribing audio input from either a file or real-time recording and providing the most probable transcription. To evaluate the effectiveness of our model's training, we utilized crucial metrics such as the optimizer (Adam), the CTC loss function, and accuracy. The CTC loss of the sequence model, as depicted in our loss plot (Fig. 5), was calculated on both the training and test datasets. A lower CTC loss value indicates a closer match between the decoded character sequence and its intended transcription. We considered additional blank tokens and redundant character duplications in the model's output, removing them after CTC beam search decoding to obtain a more understandable prediction of the character sequence. Furthermore, we quantified the model's performance using the Character Error Rate (CER) metric on the test dataset, measuring the rate of incorrect character predictions. Our model, a combination of ResNet, 1D-CNN, and BiLSTM with 1.55 million parameters, was trained for 47 epochs. With an optimal epoch of 46, we achieved a CTC of 17.98%, corresponding to an 82.02%-character accuracy rate, on the unseen test dataset. These metrics demonstrate the effectiveness of our model in accurately transcribing audio input.

We also employed a bidirectional LSTM architecture to process speech features as input, with the bidirectional LSTM layers capturing temporal dependencies in the speech. Following these layers, dense layers and a SoftMax layer predict the probability distribution of words or characters in the speech input. The model is lightweight with 1.1 million parameters, ensuring efficiency and reduced resource consumption. It consists of around 7 layers, striking a balance between complexity and performance. The model was trained for 35 epochs, achieving the desired outcomes in speech transcription tasks while maintaining a streamlined structure. We evaluated the model's performance using the Character Error Rate (CER), a metric that measures the rate of incorrect character predictions in the transcription. Our model achieved a CER of 23.69%, indicating the proportion of characters that were incorrectly predicted in the transcriptions. This CER value highlights the effectiveness of our model in transcribing speech.

A transformer model was employed to process speech data and convert it into text. This model utilizes advanced self-attention mechanisms to capture and understand the sequence and context of the speech. The workflow includes text preprocessing and embedding layers, followed by transformer encoder and decoder layers. As a deep neural network, it excels at capturing long-range dependencies in speech data. With 4 million parameters, the model strikes a balance between complexity and performance. Trained for 30 epochs, it achieved a Character Error Rate (CER) of 22.72%, indicating its effectiveness in accurately transcribing speech.

For hyperparameter tuning, we experimented with several key configurations to optimize model performance. The learning rate, which controls the size of the model's weight updates during training, was tested over a range of values between 0.001 and 0.0001. We found that a learning rate of 0.0005 provides the best balance and ensures stable convergence without overshoot or undershoot. In addition, the batch size, which determines the number of training samples processed before updating the model parameters, was tuned between 32 and 64. This allowed us to balance computational efficiency and model accuracy. The model architecture was also modified by changing the number of layers in BiLSTM and CNN, testing different combinations to find the most efficient configuration. We used the Adam optimizer, known for its adaptive learning speed and ability to handle sparse transitions, which helped speed up convergence. Training took place in 47 epochs, with optimal performance achieved after 46 epochs, as additional epochs did not lead to significant improvements. For evaluation, we used Character Error Rate (CER) as the primary metric to assess transcription accuracy. CER measures the proportion of incorrect character predictions and provides a fine-grained assessment of how well the model transcribed the speech input. In addition, the Temporal Connection Classification (CTC) loss feature was used to solve the alignment problem between the input audio and the corresponding text output. CTC is well-suited for cross-sequence problems where the input and output lengths may differ, and helped refine the alignment of predicted transcripts with real speech.

The 1D-CNN+ResNet+BiLSTM model performed better than both the Transformer and the BiLSTM models because it handled speech data more completely. The 1D-CNN helped the model pick up on small details in speech, like sounds and how they change over time, which are important for accurate transcription. ResNet was able to extract important patterns from the speech spectrograms without losing information, making the model better at handling challenges like different accents or background noise. The BiLSTM then processed the speech sequence by looking at both past and future information, improving how well the model understood the context of the speech. Together, these parts worked more effectively than the Transformer, which is good at capturing long-term connections in speech but doesn't extract local details as well as the CNN and ResNet layers. The BiLSTM-only model, while able to process sequences, lacked the feature extraction power provided by CNN and ResNet, making it less effective. As a result, the 1D-CNN+ResNet+BiLSTM model had a much lower error rate, making it the most accurate option for transcribing speech in this comparison.Table (1) evaluates multiple model combinations on the dataset to assess their performance in transcription accuracy. The models tested include 1D-CNN+RESNET+BiLSTM, Transformer CNN, and BiLSTM. Among these, the combination of 1D-CNN+RESNET+BiLSTM consistently produced the best results, achieving a Character Error Rate (CER) of 17.06%, which is the lowest CER among the three models tested.Table (2) highlights the predicted transcriptions generated by each model for a variety of actual transcriptions. The results show that 1D-CNN+RESNET+BiLSTM produced transcriptions that closely match the actual text, outperforming both the Transformer CNN and BiLSTM models. The errors observed in the 1D-CNN+RESNET+BiLSTM model are relatively minor compared to the more pronounced inaccuracies in the other two models. This suggests that the residual connection and bidirectional LSTM layers in the hybrid architecture effectively capture both local and contextual information, leading to more accurate predictions.In contrast, the Transformer CNN and BiLSTM models struggled more with complex sentence structures, often missing keywords or generating irrelevant predictions. Thus, Table (2) provides strong evidence that the 1D-CNN+RESNET+BiLSTM combination is better suited for this transcription task, as it delivers the most accurate and reliable results.

6. Conclusion and Future Enhancements

We have trained several models and discovered that ResNet combined with 1D-CNN and BiLSTM produces the optimal result. Because of the efficient data-cleaning process, the alignment between the audio frames and their corresponding characters is improved. Furthermore, we can enhance this system by using diverse datasets collected through organic methods, employing sophisticated machine learning algorithms to build the query system and produce tailored outputs, and conducting more research on live speech-to-text conversion. While our study builds upon established architectures such as 1D-CNN, ResNet, and BiLSTM, the novelty lies in the integration and optimization of these techniques specifically for Nepali speech recognition, an under-researched language in the ASR domain. By carefully fine-tuning these models with

Nepali-specific datasets and employing advanced preprocessing techniques like silent gap clipping and numeric transcript removal, we achieved a significant improvement in Character Error Rate (CER). This combination of models, alongside our novel preprocessing strategies, represents an important step forward in the development of more effective ASR systems for under-resourced languages.

Acknowledgments

We express our gratitude towards the Department of Electronics and Computer Engineering, Kantipur Engineering College, for helping us conduct the research as an academic project. We convey our profound appreciation to Er. Bishal Thapa for providing us with his support and the timely checks of our models.

References

Bhatta, B., Joshi, B. & Maharjhan, R., 2020. *Nepali Speech Recognition Using CNN, GRU and CTC.* Taipei, Taiwan, Proceedings of the 32nd Conference on Computational Linguistics and Speech Processing (ROCLING 2020).

Falyo, D. & Holland, B., 2017. *Medical and psychosocial aspects of chronic illness and disability.* s.l., s.n.

Kaldi ASR, 2011. *Kaldi Speech Recognition Toolkit.* [Online]
Available at: https://github.com/kaldi-asr/kaldi
[Accessed 2023].

Kjartansson, O. et al., 2018. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In: *Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018).* s.l.:s.n., pp. 52-55.

Muda, L., Begam, M. & I., E., 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *Journal of Computing,* 2(3).

Skyler, J. et al., 2017. *Differentiation of diabetes by pathophysiology, natural history, and prognosis.* s.l., s.n.

Sodimana, K. et al., 2018. A Step-by-Step Process for Building TTS Voices Using Open Source Data and Frameworks for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese. In: *Proc. 6th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2018).* s.l.:s.n., pp. 66-70.

Tibet Himalaya Initiative, 2006. *Tibet himalaya initiative.* [Online]
Available at: https://www.colorado.edu/tibethimalayainitiative/Resources-and-Partners
[Accessed 2023].