**Journal of the Institute of Engineering**
Volume 17, No. 1
ISSN: 1810-3383
Published: April 2023
© TUTA | IOE | PCU
Printed in Nepal

# End to End based Nepali Speech Recognition System

Basanta Joshi [a], Bharat Bhatta [b], Ram Krishna Maharjan [c]

[a,c] *Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Nepal*

[b]*Department of Electronics and Computer Engineering, Sagarmatha Engineering College, Institute of Engineering, Tribhuvan University, Nepal*

Corresponding Author: [a]basanta@ioe.edu.np,
[b]bharat.bhatta@sagarmatha.edu.np

Received: 2023-02-17 Revised:2023-03-27 Accepted: 2023-04-03

**Abstract:**
Today, technology is an indispensable part of life. To make familiar with the technology, Automatics Speech Recognition (ASR) system plays an important role. For the Nepali language due to inadequate spoken corpus, there has not been much research work, and there is not such a good model that can perform ASR. This paper presents an idea for constructing the end-to-end based Nepali ASR system and the necessary data (spoken corpus) for the Nepali language. The Nepali ASR system is able to translate spoken Nepali language to its correct textual representation. The system is built using the MFCC feature extraction, CNN for spatial feature extraction, GRU to construct the acoustic model, and CTC for decoding. The best model is built by using tuning the batch size and varying the number of the GRU units and GRU layers. This model (without using language model) provides the WER of 49.85%, 46.39%, and 52.89% on the train, validation, and test data respectively. And by using the uni-gram language model, the final model provides the WER of 35.40%, 37.50%, and 39.72% on train, validation, and test data respectively.

**Keywords**: Nepali speech recognition, Automatic speech recognition, End to end speech recognition, Gated recurrent unit (GRU), Convolution neural network (CNN)

## 1.Introduction

Speech has been a fundamental form of communication since human civilization has begun. Speech recognition could help in recognizing human speech and secondly communicating with the computers so as to ease the interactions of humans with computers. Automatic Speech recognition (ASR) systems have been developed to convert the raw audio signal to its textual transcriptions. Till now for the Nepali language, few papers have been published in speech recognition and speech synthesis. Speech recognition involves the conversion of voice-to-text representation and speech synthesis involves the conversion of text-to-speech. Nepali text-to-speech synthesis system is developed by using the concatenative approach employing the Epoch Synchronous Non-Overlap Add Method (ESNOLA) [1]. Nepali speech recognition systems are developed by using HMM (Hidden Markov Model) [2] and deep learning-based models [3, 4, 5]. In the context of the English Language or other high-resource languages, the system gets trained on a large dataset so that the generalization capabilities of the model get increased. Thus, building a dataset will help to enhance more research work in Nepali speech recognition. So, the major contribution that has been done in this research work is building the dataset (Nepali Spoken Corpus) and building an appropriate model that can predict Nepali transcription. The accuracy of the previous model is not satisfactory. This research work provides a model that can translate the spoken Nepali language to its textual representation.

## 2. Literature Review

There have been numerous researches carried out in the sector of automatic speech recognition in the

English Language. ASR starting from the digit recognizer for a single speaker using the formant frequencies measured/estimated during vowel regions of each digit reaches to Hidden Markov Model (HMM) based speaker independent ASR [6]. With the evolution of deep learning, Deep Neural Network (DNN) domination in ASR started, which showed that feed-forward DNN outperforms (Gaussian Mixture Model) GMM in the task of estimation of context-dependent HMM state emitting probabilities [7]. Till now for the Nepali language, few papers have been published in speech recognition and speech synthesis. Speech recognition involves the conversion of voice-to-text representation and speech synthesis involves the conversion of text-to-speech. Nepali text-to-speech synthesis system is developed by using the concatenative approach employing the Epoch Synchronous Non-Overlap Add Method (ESNOLA) [1]. On Nepali speech recognition systems, few research works have been published. One of the first papers based on HMM (Hidden Markov Model) for speaker-independent isolated word ASR system for the Nepali Language [2] using the six-state Hidden Markov Model (HMM). Although it has established one of the milestones in the development of Nepali ASR, it used a recording of isolated words with limited speakers and the overall accuracy of the presented system is about 75%. A Neural Network based Nepali Speech Recognition model, RNN (Recurrent Neural Networks) is used for processing sequential audio data. CTC is used as a probabilistic approach for maximizing the occurrence probability of the desired labels from RNN output. After processing through RNN and CTC layers, Nepali text is obtained as output. On implementing a trained model, audio features are processed by RNN and Softmax layer successively. The output from the Softmax layer is the occurrence probabilities of different characters at different time steps. The task of decoding is to find a label with maximum occurrence probability [8]. The model could not predict the labels that occur very close together suggesting that the network learned them as single sounds although they are multiple characters. The size of the neural network and dataset must be increased so that the network can learn accurately. These systems need some modification to increase the performance of the model. The system must get trained on a large dataset so that the generalization capabilities of the model get increased. Thus, building dataset will help to enhance more research work in Nepali speech recognition. This paper discusses two major things: building the dataset for the Nepali language and building the model using this dataset. A dataset of the duration of 23.5 hours has been prepared. The voice of 25 male speakers and 27 female speakers has been recorded. Using this dataset, a model is built on a series of steps. The performance of the model is tested for different batch sizes, varying numbers of (Gated Recurrent Unit) GRU on each layer, changing the filter size of CNN, and comparing the features MFCC and delta square MFCC. To improve the performance of the model unigram language model is employed along with the spelling corrector.

## 3. Methodology

The architecture of our deep neural network model is illustrated in Figure 1. The model receives audio as an input and its feature gets extracted, then passed through CNN block, and each of these gets convoluted and later batch is normalized. CNN blocks as followed by the two GRU network and a fully connected layer. CTC is responsible for decoding and gives the output sequence Y. Given an acoustic input X = {$x_1$, $x_2$, $\overrightarrow{x_3}$,,, $x_T$ } of length T which have the label sequences L = {$l_1$,L−2, $\overrightarrow{L_3}$,,,,,,$L_N$}. These labels are Nepali characters, letters, or words of length N.

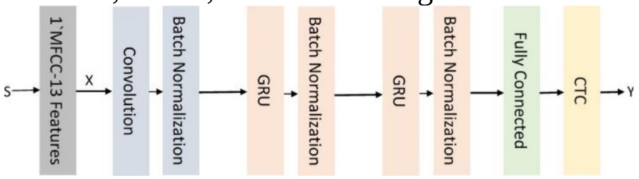

**Figure 1:** Architecture of model for speech recognition

The inputvector $x_t ε R^D$ represents D dimensional input speech vector of the Mel-filter bank corresponding to $t^{th}$ speech frame. If V is labels vocabulary, $l_u εV$ is the label at position u in L, then $V^*$ represents the collection of all label sequences formed by labels in V. From this information, the

ASR needs to find the most likely label sequence $\hat{L}$ for given X[9]. This can be represented by Equation 1:

$$\hat{L} \; = \; arg_{L\varepsilon V^*}^{max} \, P(L/X) \qquad \ldots(1)$$

The main objective of an ASR is to establish a model that can accurately calculate the posterior probability $p(L/X\,)$. $P(L/X\,)$ is the probability of occurrence of label $L$ for given input feature vector $\vec{X}$.

## 3.1 Feature Extraction

For Speech feature extraction, speech files have been segmented into a fixed-sized window of about 20 ms and MFCC (Mel-frequency cepstral coefficients) have been extracted [10]. These audio features extracted are sent to CNN.

## 3.2 Feature Learning using CNN

The plot of MFCC displays a transformed intensity of frequencies over time and closely resembles natural images[11], hence CNN can be used to capture high-level features in the spatial domain. 1-dimensional CNN of filter size 400, kernel size (K) 11, and dilation 1 was used. The output of CNN is given by the equation

$$(I * K)_i = \sum_{-\infty}^{\infty} I_\eta .\; K_{\eta-1} = \sum_{-\infty}^{\infty} I_{\eta-1} .\; K_\eta \qquad \ldots(2)$$

After performing the convolution another operation, known as maxpool is carried out. The maxpool is carried out by taking the pool size=2 and strides=1. Then, batch normalization is carried out by the relation is from [12].

$$\hat{X}^{(k)} = \frac{x^k - E[x^k]}{\sqrt{var[x^k]}} \qquad \ldots(3)$$

## 3.3 GRU Acoustic Model

Input from the CNN after batch normalization is sent to the input of the GRU network[13]. The output is again normalized and fed to the sequence of GRU and batch normalization. This output is passed into the softmax function. The posterior probability of each output layer (that represents 93 symbols) is computed and the unit (symbol) having the highest probability gives the output. The decoding is carried out by the CTC. The input to CTC is supplied from the output of the softmax. The decoded output of CTC is altered and mapped back to the Nepali language character. This mapped output is the predicted output of the model. The loss is computed by comparing the true labels and predicted labels. Later the model weights get adjusted.

## 3.4 Dictionary

The dictionary is used to correct the word that goes on the wrong prediction. To build the dictionary, the Nepali text corpus is collected and it gets broken down into words. Each word is stored in the dictionary along with its word count. The counts represent the number of times it gets appeared in the sentences. Uni gram language model is applied. Which will further correct the predicted model output.

## 4. Dataset

Speech corpus is a database of speech audio files and text transcriptions of them, in a format that can be used to create Acoustical Models using speech recognition engines. Speech recognition system involves in the supervised learning process. Hence to build the system labeled data is needed[14].

## 4.1 Text Corpus

The preliminary requirement to build the Nepali spoken corpus is plain Nepali text. The plain Nepali text is collected by web mining of sports news from different online news portals: Setopati, Annapurnapost, and Kantipur sports news using python module: beautifulsoup is used for the mining process. News under the category of sports, subcategory:South-Asian-Games, cricket, and football are extracted and kept in text file[15]. This text file consists of 423302 words.
The text corpus contains different tokens and punctuation. Tokenization is carried out. These punctuations are removed and some tokens needed to be replaced. The punctuation signs ”,!,?,”,’,@,,, are

removed form the text corpus. Some of the Nepali digits १,२,३,१२ .. etc get replaced with their number name एक , दुइ, तीन्, बाह्... etc respectively. The mined text contains some tokens (hyperlinks, some English words, and digits) which get removed from the text file. Number of text file are created. The corpus present in the text file will be recorded as an audio file. Each text corpus file consists of 200 sentences. Each sentence is formed by a combination of 10 words. These text corpora are spoken by the speaker and recorded to form an audio file. A single text file (label file) consists of 200 sentences and hence 200 audio voices are recorded for a single speaker.

## 4.2 Audio Corpus

A single-text corpus file (label file) consists of 200 sentences. The speaker speaks each sentence in the text corpus file and it is recorded. For a single text corpus file, there are 200 audio voice files. The voice file is recorded in wav format. The overall recorded data has a duration of 23.5 hours. The recording is carried out on android smartphones by using the application named ”Voice Recorder”. An audio file is recorded in .wav file format in CD quality i.e sampling is carried out on 44100 kHz.

### 4.2.1 Speaker Distribution

The speakers get selected on the basis of their age and on the basis of gender. 25 Male speakers and 27 female speakers have contributed to building the speech corpus. We have tried to make the same number of speakers in each case during the distribution of speakers. The distribution of speakers on the basis of age is shown in the table The speakers are selected from the students, staff, and teachers of Sagarmatha Engineering College. A text corpus is provided to speakers and instructed to record them.

**Table 1:** Speaker distribution on the basis of age.

| Age group | Numbers of speaker |
|---|---|
| 16<age<20 | 10 |
| 20<age<30 | 32 |
| age>30 | 10 |

### 4.2.2 Recording Device and Application

The speakers recorded the audio corpus by using their smartphones. An android application named “Voice Recorder” is used to record the text corpus[16]. The recording is carried out in 44.1 kHz.

## 4.3 Corpus Cleaning

The audio corpus and text data (label) must be aligned together. A manual check is carried out by listening to each recorded audio file and getting checked with their corresponding label (text) and then correction is carried out if necessary. The major correction along with the problem is discussed:

- Mispronounced word: Mispronounced word gets corrected in text corpus according to speaker pronunciation if it gives appropriate meaning else both the audio file and its textual representation get removed.
- Speaker noise: Audio file in which sounds made by a speaker other than words, like clearing the throat or exhaling get removed.
- Background noise: Audio files having high-intensity background noise like vehicle horn sound, the sound of a door slamming, and vehicle movement noise get removed.
- Split audio corpus: An audio corpus having a length greater than 12 seconds get split into two parts. And then their label (text) gets aligned.

## 5. Experiment

The experiment is carried out on two different platforms. One was carried out on the Nvidia GeForce MX150 GPU and another is carried out on google collab. A series of experiments have been conducted on this setup to obtain the best model. The model is prepared in the python programming language and its library.

## 5.1 Data Augmentation

To perform Deep Learning, the model needs a large amount of data to build the models that can well predict both seen and unseen data. Even with a small dataset Deep learning can be carried out by using Data Augmentation (DA). DA increases the quality of training data[17]. The augmentation techniques used are random noise addition, time shift, and time stretch. A random amount of Gaussian noise is superimposed in the original signal. Random noise is generated and then 0.5% of this generated noise is added to the original signal. Following a similar technique from [18], the whole process is to enhance the robustness of the system operation even in a noisy environment. The idea of shifting time is quite simple. It just shifts the audio to left/right with a random second. The signal is shifted by 8000 steps towards the rights[19]. Time stretch, stretches times series by a fixed rate. It slows down or speeds up the audio. A stretching factor of less than 1 speed up the sample and a stretching factor greater than 1 slow down the audio sample keeping the pitch unchanged[20]. The time is stretched by factors of 0.8 and 1.2.

### 5.2 Data Split

To train the model the dataset needs to split into train, test, and validation. For the final model, data is split into train, test, and validation and for other experiments, the dataset is split into train and test. For the final model train, the test and validation split is 60%, 20%, and 20% respectively. And for the other experiments train data and test data are split into 80% and 20% respectively.

### 5.3 Evaluation

Word Error Rate (WER) is a common metric used to compare the accuracy of the transcripts produced by speech recognition. The word error rate can then be represented mathematically as:

$$WER = \frac{S+D+I}{N} \qquad \ldots(4)$$

Where,
S represents no. of words substituted,
D represents no. of words deletions,
I represents no. of words inserted,
C represents no. of words correctly predicted,
N represents no. of words in ground truth,
(N = S+D+C)

### 5.4 Training

The objective of the training is to find out the best model for the training. The model is trained by changing different hyperparameters. The details of the experiments have been discussed in the following section.

## 6. Results and Discussion

The first experiment is carried out to find out the appropriate number of GRU layers needed in the model. The architecture of the first model is CNN followed by a single GRU layer, the second model consists of CNN followed by two GRU layers and the third model consists of CNN followed by three GRU layers. The number of GRU units in each layer is 200 with a value of dropout of 0.5. The CNN kernel size is 200 and the filter size is 11. The MFCC has dimensions 0f 13, windows size of 25ms, and filterbank size of 26. The mini-batch size is 5. The SGD (Standard Gradient Descent) is used to train the model. The WER for the first model, second model, and third model is 90%, 82%, and 57.5% respectively. The next experiment is carried out to find the appropriate batch size. An experiment on batch sizes 5, 15, and 35. The model architecture is CNN followed by two GRU units. The other parameters remain unchanged. The WER for batch sizes 5, 15, and 35 are 67%, 68%, and 80.5% respectively. With an increase in the batch size, the train-test WER combination relatively increases thereby reducing the generalizing capabilities. So, lower batch size is preferred in this scenario.

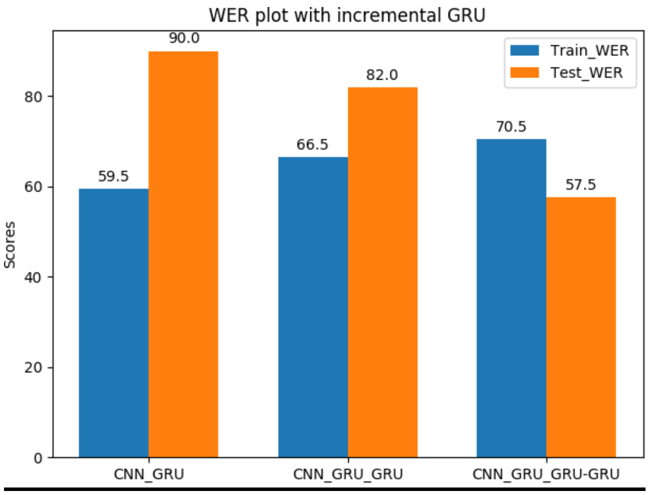

**Figure 2:** WER plot on changing GRU layer

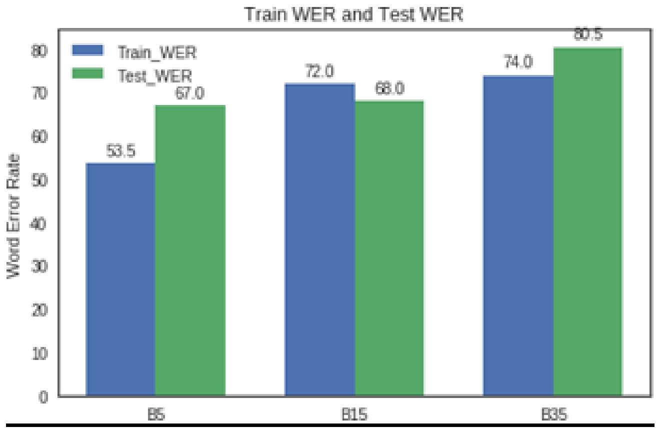

**Figure 3:** WER plot on changing batch size

Another experiment is carried out for two different kernel sizes: 128 and 220. The WER for validation and test data on filter size 128 are 54.83 and 57.8 respectively and WER for validation and test data on filter size 220 are 51.62% and 52.15% respectively. An experiment is carried out to study the effect on WER by changing the units of GRU in each GRU layer. The study is carried out by changing numbers of unit size from 200 to 400. With an increase in number of units of GRU in each GRU layer, the WER testing data is decreased. The WER for 200 units and 400 units are 71% to 67% respectively

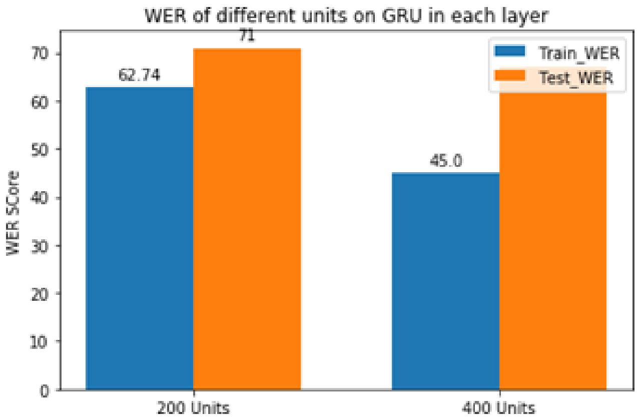

**Figure 4:** WER plot on changing units of GRU in each layer

Since all the model that has been built before does not give the best prediction. One of the reasons is to go on data augmentation. After augmentation duration of the spoken corpus is increased from 23.5 hours to around 80 hours. The batch size gets increased up to 100. The WER for the validation and test data is found to be 46.85% and 52.89%. The final model consists of a CNN layer followed by two GRU units. There are 400 units of GRU in each layer with a dropout of 0.5. The CNN kernel size and filter size are 220 and 11 respectively. The MFCC has dimensions of 13, a windows size of 25ms, and a filterbank size of 26. To increase theaccuracy of the model word corrector and uni-gram language model is implemented.

**Table 2:** Model building by using augmented data

| Model | Test WER | Valid WER |
|---|---|---|
| Without Dictionary | 49.39% | 52.89% |
| With Dictionary | 37.5% | 39.72% |

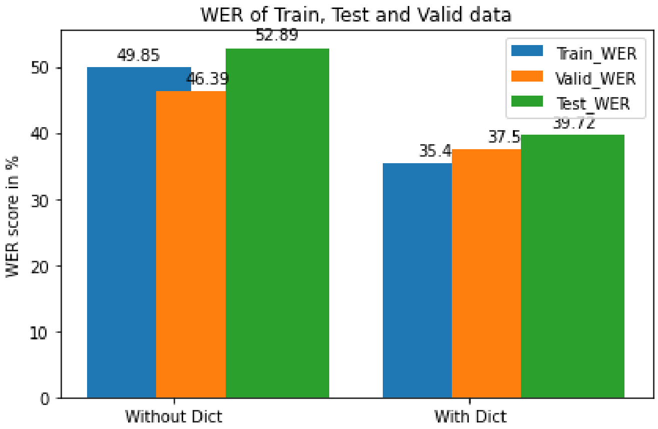

**Figure 5:** WER plot with dictionary and without dictionary

Table 2 shows the result of the final model. With the increase in the number of GRU layers, the WER of the model decreases but the training duration of the model increases. With the increase in the size of batch size, the WER increases. This shows that generalizing capabilities of the model increased with a decrease in batch size. The WER is found less when the kernel size of CNN gets increased. The WER also decreased with an increase in the number of GRU units in each layer. By taking the above information a final model is built by the data augmentation. The WER of the model gets decreased by using the data augmentation technique. Finally, a dictionary and word corrector is employed to reduce the WER. With the implementation of the dictionary, the testing accuracy is changed by 25%.

## 7. Conclusion

End-to-end based Nepali ASR involves translating the Nepali language to its textual representation. This research works provides an idea for building Nepali speech recognition from scratch by using deep learning algorithms. This research work studies the effect of different hyperparameters on the ASR system and using the knowledge from the several conducted experiments, a final model is created that can perform ASR with less WER. The research work provides an idea for building the spoken corpus. Thus, this built spoken corpus is used to train the Nepali ASR system. Further, the accuracy of the model gets increased by 25% when using the language model.

## Acknowledgements

This work has been supported by the University Grants Commission, Nepal under a Faculty Research Grant (UGC Award No. FRG-76/77- Engg-1) for the research project ”Preparation of Nepali Speech Corpus: Step towards Efficient Nepali Speech Processing.

## References

[1] B. Chettri and K. B. Shah, “Nepali text to speech synthesis system using esnola method of concatenation,” International Journal of Computer Applications, vol. 62, no. 2, 2013.

[2] M. K. Ssarma, A. Gajurel, A. Pokhrel, and B. Joshi, “Hmm based isolated word nepali speech recognition,” in 2017 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1. IEEE, 2017, pp. 71–76.

[3] B. Bhatta, B. Joshi, and R. K. Maharjhan, “Nepali speech recognition using cnn, gru and ctc,” in Proceedings of the 32nd Conference on Computational Linguistics and Speech Processing ({ROCLING} 2020), 2020, pp. 238–246.

[4] J. Banjara, K. R. Mishra, J. Rathi, K. Karki, and S. Shakya, “Nepali speech recognition using cnn and sequence models,” in 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT). IEEE, 2020, pp. 1–5.

[5] M. Dhakal, A. Chhetri, A. K. Gupta, Lamichhane, S. Pandey, and S. Shakya, “Automatic speech recognition for the nepali language using cnn, bidirectional lstm and resnet,” in 2022 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2022, pp. 515–521.

[6] S. Furui, “50 years of progress in speech and speaker recognition research,” ECTI Transactions on Computer and Information Technology (ECTI- CIT), vol. 1, no. 2, pp. 64–74, 2005.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, Nguyen, T. N. Sainath et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal processing magazine, vol. 29, no. 6, pp. 82–97, 2012.

[8] P. Regmi, A. Dahal, and B. Joshi, “Nepali speech recognition using rnn-ctc model,” International Journal of Computer Applications, vol. 178, no. 31, pp. 1–6, Jul 2019.

[9] D. Wang, X. Wang, and S. Lv, “An overview of

end- to-end automatic speech recognition," Symmetry, vol. 11, no. 8, p. 1018, 2019.

[10] B. Joshi, B. Bhatta, S. P. Panday, and R. K. Maharjan, "A novel deep learning based nepali speech recognition," in Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2022, Volume 2. Springer, 2022, pp. 433–443.

[11] W. Song and J. Cai, "End-to-end deep neural network for automatic speech recognition," Standford CS224D Reports, 2015.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[13] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 2, pp. 92–102, 2018.

[14] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in International conference on machine learning, 2016, pp. 173–182.

[15] "Nepal's digital newspaper :: :: Setopati," https://www.setopati.com/sports/ South-Asian-Games, (Accessed on 07/13/2020).

[16] "Voice recorder - apps on google play," https://play.google.com/store/apps/details?id=com.media.bestrecorder.audiorecorder&hl=en , (Accessed on 07/13/2020).

[17] I. Rebai, Y. BenAyed, W. Mahdi, and J.-P. Lorré, "Improving speech recognition using data augmentation and acoustic model fusion," Procedia Computer Science, vol. 112, pp. 316–322, 2017.

[18] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates et al., "Deep speech: Scaling up end-to-end speech recognition," arXiv preprint arXiv:1412.5567, 2014.

[19] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," arXiv preprint arXiv:1904.08779, 2019.

[20] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279–283,2017.