

BanglaNum - A Public Dataset for Bengali Digit Recognition from Speech

Mir Sayeed Mohammad, Azizul Zahid, Md Asif Iqbal

Department of Electrical and Electronic Engineering

Bangladesh University of Engineering and Technology

Abstract—Automatic speech recognition (ASR) converts the human voice into readily understandable and categorized text or words. Although Bengali is one of the most widely spoken languages in the world, there have been very few studies on Bengali ASR, particularly on Bangladeshi-accented Bengali. In this study, audio recordings of spoken digits (0-9) from university students were used to create a Bengali speech digits dataset that may be employed to train artificial neural networks for voice-based digital input systems. This paper also compares the Bengali digit recognition accuracy of several Convolutional Neural Networks (CNNs) using spectrograms and shows that a test accuracy of 98.23% is achievable using parameter-efficient models such as SqueezeNet on our dataset.

Index Terms—spectrogram, convolutional neural networks, short-time fourier transform, window function

I. INTRODUCTION

Speech recognition is an essential part of human-computer interaction that can facilitate convenient ways to work with intelligent machines. Nowadays, many computerized systems such as digital voice assistants, voice-based input systems, task-specific chat-bots on e-commerce sites or customer care services, personal assistants, etc benefit from speech recognition and can interact with the users naturally. These systems have been researched for a long time, particularly for languages like English, Chinese, etc. Modern speech recognition systems utilize artificial neural networks, and their performance is largely dependent on the availability and volume of accurately labeled natural language datasets. In the case of Bengali, speech datasets are less available which impedes the research and development of intelligent speech recognition systems that can understand Bengali. Instead of complex speech systems, digit recognition can simplify the task of building a voice-based automatic system. Thus to build a simple voice-based user interface using Bengali digits, we aimed to create a language dataset and develop a speech recognition system based on it. Consequently, we first collected speech data, preprocessed it, and then trained a neural network architecture for automatic speech recognition based on it. This paper is organized as follows - we first discuss the existing works on Bengali speech recognition. Then we describe our data collection and post-processing procedures in detail. Following that, we present experimental classification results using a convolutional neural network on the speech spectrogram of digit utterances from our dataset. Finally,

we highlight the limitations of our work and discuss future research directions based on it.

II. RELATED WORK

Research on speech recognition has been active since the 1930s and many studies have been published in the most widely spoken languages such as English [1], Chinese [2], Indian [3] and Portuguese [4]. But studies to recognize Bengali speech have only begun in recent times. Here we have compiled the most up-to-date works on Bengali speech recognition problems that scholars have attempted to solve using various approaches. In [5], researchers created a Bengali Number Recognition (BNR) system based on Convolutional Neural Networks (CNNs). Their proposed algorithm correctly distinguishes Bengali spoken numbers (0-99) with an accuracy of 89.61%. Besides, different Gaussian Mixture Model-Hidden Markov Models (GMM-HMM) have been explored in [6] to develop a voice search module for the search engine "Pipilika". In [7], an alphabet of 39 phoneme symbols has been devised to categorize the data more precisely using Multi-layer Perceptron Classifier (MLPC) and Support Vector Machine (SVM). A BNR system from speech input using a CNN is built in [8]. Here, a speech dataset consisting of 6000 utterances of 10 isolated Bengali digits has been introduced. A Bengali isolated spoken numerals recognition system is created in [9] that uses Mel Frequency Cepstrum Coefficients (MFCC) and GMM features. The work in [10] suggests digit recognition from a mix of Bengali and English speech. In this work, the authors used an open-source dataset for English and created a new Bengali dataset in noisy environments from speakers of various ages, gender, and dialects. Then they used MFCC to extract features from the mixed dataset and a CNN classifier to train, test, and analyze the data. In [11], Short-Time Fourier Transform (STFT) is employed to create feature vectors in an HMM-based isolated BNR system. Also, a deep learning strategy for categorizing Bengali spoken digits is proposed in [12]. It takes into account all aspects such as dialects, gender, and age groups. In our study, we propose a novel dataset for the recognition of Bengali digits from speech and follow the classification approach that was used to detect English language commands in the "Speech Commands" [13] dataset.

III. DATASET PREPARATION

The main purpose of this work is to introduce a dataset for detecting utterances of unit digits in the Bengali language

¹<https://www.kaggle.com/datasets/mirsayeed/banglanum-bengali-number-recognition-from-voice>

that can be used in automated answering systems or voice-based digital assistants on e-commerce sites. In this section, we focus on the procedures followed during the preparation of this dataset.

A. Data Collection

The data collection process was done inside a classroom environment, ensuring realistic ambient noise and minimal interference from other people talking. We have used MATLAB for recording the audio on a laptop using the microphone of generic headsets that people use for everyday work. This is also done to ensure that the dataset represents audio samples from everyday life and is easily deployable without requiring additional hardware. For generating the audio samples, native Bengali-speaking undergraduate student volunteers were asked to talk into the microphone for a duration of 40 seconds. During that time, the students had to cycle through the digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 for five times at varying rates, and then utter their student ID at a natural speed, all in Bengali. We have managed to collect audio samples from 40 people in total, with 35 male voice samples and 5 female voice samples. In accordance with the sample rate used in the work of P. Warden [13], we have used a sampling rate of 16kHz and 24 bits of sample depth.

TABLE I
DATASET DESCRIPTION

Digit	0	1	2	3	4	5	6	7	8	9	Total
Count	280	246	214	221	218	206	269	210	194	194	2252

B. Data Processing

After collecting the audio samples, we had to convert them into a more usable format for training neural network models. Each utterance of a Bengali digit was cropped out from the full 40 second audio sample manually in MATLAB. Then all of these were brought to a uniform length of 8192 samples (2^{13}) per utterance by either cropping out extra data or by zero padding on both sides. This resulted in a dataset consisting of audio samples lasting 0.512 seconds each. After cutting out the audio samples, they were saved in the *wav* format and sorted according to class labels into different folders as available in the dataset. The final dataset consists of a total of 2252 utterances of the 10 unit digits. table I summarizes the sample count in each class.

IV. CLASSIFICATION EXPERIMENTS

Experimental deployment of the dataset as a voice-based digital assistant required creating a speech command recognition model. For this purpose, we followed the method described in the work of P. Warden [13] where the voice spectrogram is fed through a simple convolutional neural network for classification. This section briefly describes the overall methodology shown in fig. 1, as well as the comparative performances of different CNN architectures and the study for hyperparameter search.

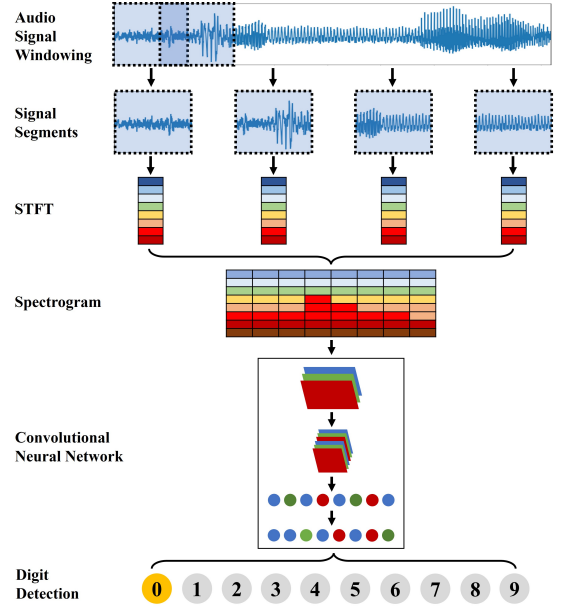


Fig. 1. Overall system architecture.

A. Calculating Spectrograms

Audio spectrogram is a very efficient and effective way of representing the important features of speech signals. It performs exceptionally well in tandem with convolutional neural networks. For this, the signal is first segmented into chunks of small time lengths. Various window functions such as rectangular, Bartlett, Hamming, Hanning, etc can be used for this, and we have used the Hanning window given by eq. (1) in our experiments. Overlaps are introduced between windows to best capture the audio features. Then, for each of these segments, the Short Time Fourier Transformation (STFT) given in eq. (2) is evaluated and the absolute square value of the STFT is taken as given in eq. (3), which represents the power in each frequency band. As a result, a 2-D array is formed in which the horizontal axis represents time, the vertical axis represents frequency, and the value of a point represents the power of the frequency component at any time. The overall process can be summarized by the following equations:

$$w[n] = 0.5(1 - \cos(2\pi \frac{n}{N})), 0 \leq n \leq N, N = L - 1. \quad (1)$$

$$\text{STFT}\{x[n]\}(m, \omega) \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-i\omega n} \quad (2)$$

$$\text{spectrogram}\{x[n]\}(m, \omega) \equiv |X(m, \omega)|^2 \quad (3)$$

Where $w[n]$ is the window function, L is the window length, $x[n]$ is the discrete time signal, $X(m, \omega)$ is the STFT of the signal.

For our work, we have used a window size of 256 and a step size of 32 (overlap of 224 samples between windows).

This gives a spectrogram with a dimension of $249 \times 129 \times 1$. Window and step sizes were determined by the hyperparameter search described in section IV-E. fig. 2 shows the waveform and spectrogram of the utterance of 0 in Bengali.

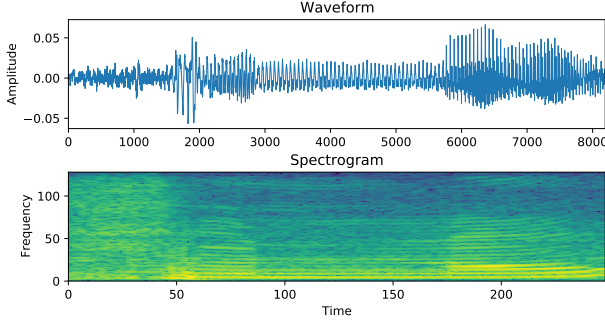


Fig. 2. Example of an audio signal and its corresponding spectrogram.

B. Convolutional Neural Networks

Spectrogram gives a visual representation of an utterance. This image can be used to classify the audio sample. For this classification task, we have experimented with different Convolutional Neural Network (CNN) architectures such as a basic custom CNN, SqueezeNet [14], EfficientNet [15] and ResNet50 [16]. Due to the small size of the dataset, we decided to use models that have a lower parameter count and do not 'remember' the training data entirely. The command recognition accuracy of these different models has been included in table II.

C. Training

For training the network, we have used 80% of the dataset, and the other 20% has been split evenly to be used as the validation and test sets. For optimization, we have used the Adam optimizer with a learning-rate of 10^{-4} and for the loss function, we have applied Sparse Categorical Cross-entropy. Each of the classification models was trained for 200 epochs with a batch size of 32. The batch size was determined by the hyperparameter search. During the training process, we monitored the training and validation losses as well as accuracy, so that we can see how our model improves over time in each successive epoch. fig. 3 shows the training and validation loss and accuracy.

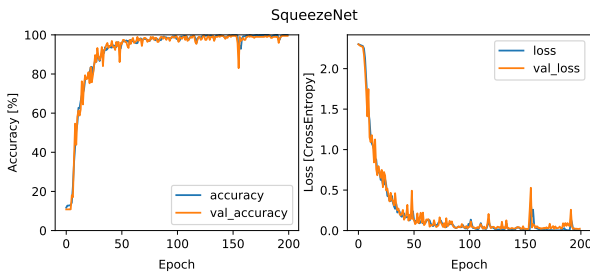


Fig. 3. Accuracy and loss plots for SqueezeNet.

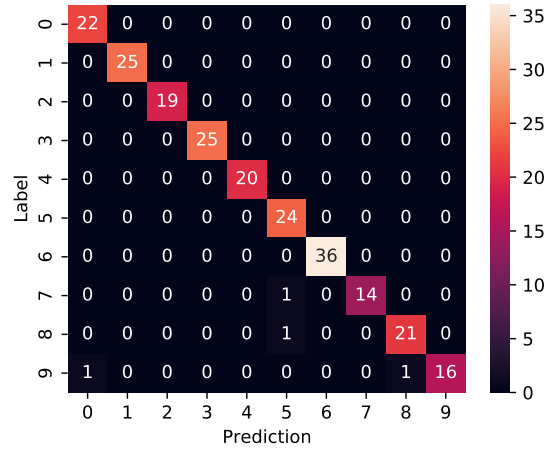


Fig. 4. Confusion matrix for test set of SqueezeNet.

D. Results and Analysis

We have tested several lightweight convolutional neural network models for training on the dataset and compared their performances. Lightweight models are preferable to avoid overfitting on the dataset as well as reduce inference times for real-time deployment. Less computational complexity enables the models to be implemented on CPU-only systems or single-board computers. So we have tested truncated versions of some of the state-of-the-art classification network architectures in our experiments. In table II we have reported the accuracies on the validation and test sets, as well as the parameter counts for each of the models.

TABLE II
BENGALI NUMBER CLASSIFICATION RESULTS

CNN Network	Val Acc.	Test Acc.	Param. Count
Basic CNN	0.9821	0.9513	61,984,525
SqueezeNet	0.9955	0.9823	726,474
EfficientNetB0	0.9598	0.9646	4,061,801
EfficientNetB0(Trunc) ¹	0.9732	0.9602	1,356,877
EfficientNetB1	0.9911	0.9735	6,587,469
EfficientNetB1(Trunc)	0.9821	0.9469	1,717,681
EfficientNetB2	0.9732	0.9425	7,782,079
EfficientNetB2(Trunc)	0.9866	0.9558	1,967,653
EfficientNetB3	0.9777	0.9602	10,798,181
EfficientNetB3(Trunc)	0.9911	0.9690	2,822,951
EfficientNetB4	0.9732	0.9292	17,690,885
EfficientNetB4(Trunc)	0.9911	0.9602	4,400,205
ResNet50	0.9911	0.9735	23,601,930

¹ In the truncated models, the model head and the last convolutional layers are removed.

In our tests, the SqueezeNet [14] architecture outperforms the other classification networks with a validation accuracy of 99.55% and a test accuracy of 98.23%. Some of the more complex architectures have lower performance compared to SqueezeNet despite having larger parameter counts. This implies that classifying Bengali digits from utterances can be accomplished using lightweight networks quite effectively. The confusion matrix of the test set is shown in fig. 4.

We can see from the confusion matrix that among the 225 samples in the test set, digits 7 (phonetic transcription: /shʌt/) and 8 (phonetic transcription: /ʌt/) are misidentified by the network as 5 (phonetic transcription: /pʌch/). This is because these numbers might sound similar when slightly mispronounced and can confuse even the native speakers. Additional processing might be required to correctly identify these two digits in Bengali speech.

E. Hyperparameter Search

In order to select the correct batch size, spectrogram window length and step durations, we have conducted an extensive hyper-parameter search using the Basic CNN and SqueezeNet models. From these tests, the parameter values for which both models performed considerably well have been selected. We have several observations from the searching process as listed below:

Batch Size: Increasing the batch size had a detrimental effect on the model performances - the training curves showed instability along with a reduction of accuracy. From a set of 256, 128, 64, 32 - the best results were obtained for a batch size of 32.

Window Size: the window size has a huge effect on the quality of information represented by the spectrogram. A large window can encapsulate a lot of frequency information but the time resolution is reduced. A smaller window loses frequency information but has better time resolution. Among window sizes of 1024, 512, 256, 128, 64 - the best result was obtained for the value of 256. This indicates that 256 samples offer optimum frequency and time resolutions at the sampling rate of 16 kHz.

Step Size: for the tested step sizes of 256, 128, 64, 32, 16 - the best step size was found to be 32. A lower step size indicates that getting spectrograms at smaller intervals can prevent loss of information due to windowing.

V. LIMITATIONS AND SCOPE FOR FUTURE WORK

In our paper, we have collected audio samples in a classroom environment. So the potential sampling bias of this dataset could be reduced by collecting data from a more diverse group of people and varying environmental conditions. Moreover, the volume of the dataset could be increased for better generalization of learning algorithms. It could also be mixed with other Bengali speech datasets to augment the speech recognition capabilities of artificial neural networks. Finally, our reported results are mainly based on a classification algorithm based on spectrograms and CNNs - this could potentially be improved by other approaches such as wavelet transform based multi-resolution analysis, recurrent neural networks or transformer based architectures.

VI. CONCLUSION

With the progress of technology, voice recognition based systems are being integrated into a variety of sectors of life. The Bengali voice recognition system is no exception. In this study, we have presented a novel dataset containing utterances

of Bengali digits. Then we have described the data acquisition process and dataset details. Furthermore, we have shown the digit identification performance using a spectrogram and CNN based classification algorithm. Given that the experimental test accuracy is 98.23% and might be further enhanced, we can infer that this dataset would be quite useful for developing Bengali voice based digital assistants for automated applications.

REFERENCES

- [1] G. S. Pawar and S. S. Morade, "Realization of hidden markov model for english digit recognition," *International Journal of Computer Applications*, vol. 98, no. 17, 2014.
- [2] S. W. Fu, C. Lee, and O. L. Clubb, "A survey on chinese speech recognition," *Communications of COLIPS*, vol. 6, no. 1, pp. 1–17, 1996.
- [3] C. Kurian, "A survey on speech recognition in indian languages," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 6169–6175, 2014.
- [4] A. de Andrade Bresolin, A. D. D. Neto, and P. J. Alsina, "Digit recognition using wavelet and svm in brazilian portuguese," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1545–1548, 2008.
- [5] O. Sen, P. Roy, and A. Mahmud, "A novel bangla spoken numerals recognition system using convolutional neural network," *Available at SSRN 4127691*, 2022.
- [6] J. Rahman Saurav, S. Amin, S. Kibria, and M. Shahidur Rahman, "Bangla speech recognition for voice search," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–4, 2018.
- [7] M. G. Hussain, M. Rahman, B. Sultana, A. Khatun, and S. A. Hasan, "Classification of bangla alphabets phoneme based on audio features using mlpc and svm," in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, pp. 1–5, 2021.
- [8] M. I. R. Shuvo, S. Akib Shahriyar, and M. A. H. Akhand, "Bangla numeral recognition from speech signal using convolutional neural network," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pp. 1–4, 2019.
- [9] B. Paul, S. Bera, R. Paul, and S. Phadikar, "Bengali spoken numerals recognition by mfcc and gmm technique," in *Advances in Electronics, Communication and Computing* (P. K. Mallick, A. K. Bhoi, G.-S. Chae, and K. Kalita, eds.), (Singapore), pp. 85–96, Springer Nature Singapore, 2021.
- [10] S. Das, M. R. Yasmin, M. Arefin, K. A. Taher, M. N. Uddin, and M. A. Rahman, "Mixed bangla-english spoken digit classification using convolutional neural network," in *Applied Intelligence and Informatics* (M. Mahmud, M. S. Kaiser, N. Kasabov, K. Iftexharuddin, and N. Zhong, eds.), (Cham), pp. 371–383, Springer International Publishing, 2021.
- [11] B. Paul, D. D. Adhikary, T. Dey, S. Guchhait, and S. Bera, "Bangla spoken numerals recognition by using hmm," in *Computational Intelligence in Pattern Recognition* (A. K. Das, J. Nayak, B. Naik, S. Dutta, and D. Pelusi, eds.), (Singapore), pp. 85–97, Springer Singapore, 2022.
- [12] R. Sharmin, S. K. Rahut, and M. R. Huq, "Bengali spoken digit classification: A deep learning approach using convolutional neural network," *Procedia Computer Science*, vol. 171, pp. 1381–1388, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).
- [13] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.
- [14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.
- [15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.