Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 1381–1388

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Bengali Spoken Digit Classification: A Deep Learning Approach Using Convolutional Neural Network

Riffat Sharmin[a], Shantanu Kumar Rahut[b] , Mohammad Rezwanul Huq[c1],

[a,b,c]East-West University, Dhaka-1212,Bangladesh

Abstract

Bengali is a largely spoken language. Bengali speech recognition can have a significant effect in many fields such as human-computer interaction, the internet of things, etc. A part of a Bengali speech recognition system is the process of Bengali spoken digit classification. A few works have been done on Bengali digit classification, but all of them had missed out on one or two influential parameters like dialects, gender or age-groups. Voice of people differs due to gender, dialects, and age. This paper proposes a deep learning approach for classifying the Bengali spoken digits. It takes all parameters like dialects, gender, age-groups into account and the proposed approach acquires more than 98% accuracy using a convolutional neural network (CNN).

Keywords: Bengali Spoken Digit classification; Convolutional Neural Network; Deep Learning

1. Introduction

When a machine or computer program can recognize phrases and words in the form of spoken language and can transform them into machine-readable form, then it is known as speech recognition [1]. As humans' language is not similar to computers' language, so there comes a necessity to convert the human language in such a way so that computers can understand it. In that case, Natural Language Processing helps us to do so. Various kinds of algorithms and acoustic and language models are being used as the working method of the speech recognition process [1]. Various types of works such as- voice recognition, voice search, voice transcription, call routing, speech

[1]* Corresponding author.
E-mail address: mrhuq@ewubd.edu

1877-0509 © 2020 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2020.04.148

to text process can be done easily by using speech recognition techniques. Numerous works have been carried out on different kinds of languages but in terms of the Bengali language, there is not that much work on it. Approximately 228 million people speak Bengali as their first or second language [2]. So, it has now become a necessity to work appropriately on the field of speech recognition in the Bengali language. As the term ‘speech' in the case of Bengali language is quite vast on its own, so the main aim of this paper is to recognize the ‘digits (0-9)' in the Bengali language.

2. Related work

Numerous works have been done with digits on various languages. Such as- English, Hindi, Malayalam, Bundelkhandi, etc. In paper [3], the authors reported Microsoft’s conversational speech recognition system for 2016. They used CNN and RNN and found that lattice-free MMI training and I-vector modeling had a notable achievement for all the model's architecture. The error rate for a single system was 6.9%. And for the combined system is 6.2 %. In [4], the authors found that LSTM performed significantly in an SNN. They were successful in recognizing any digit with 88% probability, and when they did more conventional pre-processing, the probability was 99.4%. In [5], the authors compared Bidirectional LSTM to various other neural network architectures. They found that compared to standard RNN and MLPs, LSTM is much faster. Also, Bidirectional networks perform better than unidirectional networks. In [6], the authors worked on Hindi digits and built an HMM-based model. They found out that, on the word level the accuracy of training data is 94.09% and on testing, data is 85%. On the phone level, accuracy over the training data is 92.82% and testing data is 86.17%. In paper [7], the authors found that a well-trained database led to a proper recognition system. Also, in each state, the system accuracy increased if the number of mixtures was increased. The work, reported in [8], is on the Bundelkhandi digit. LPC and Modified MFCC algorithm were used in this work. From the usual MFCC and LPC algorithm, the proposed algorithms by the author performed well. Paper [9] is based upon the Pashto language. They used MFCC for feature extraction. And for classification, KNN was used. The accuracy of this model was 76.8%.

In terms of the Bengali language, the works on digits are not that wide. Few works can be found in the case of digit recognition in the Bengali language. In paper [10], the authors used HMM and MFCCs. They discovered that digits from 0-5 showed higher accuracy (more than 95%) and digits from 6-9 showed lower accuracy (less than 90%). Moreover, they found that because of different dialects the two pairs of digit- 6(৬) and 9(৯), 7(৭) and 8(৮) were confusing. The authors in paper [11], the authors conducted a HMM-based back end along with a front-end based Mel-LPC. The accuracy of this dataset was 98.05%. In article [12], the authors used three types of approaches were used in this experiment. Among these three models, MFCC one gave better performance. But the drawback of the MFCC model was that it had faced the over-fitting problem. In paper [13] The authors used MFCC for feature extraction, Vector quantization to lessen the dimension and to generate a vector codebook. DTW was used as a classification based and also used a minimum distance classifier. The authors found that MFCC based speech processing approach showed some limitations in the presence of noise for speaker-independent spoken digit recognition scenarios. In [14], authors have targeted the Bengali language. In this paper, they worked with MFCC features with PCA for dimension reduction. They used support vector machine, random forest and multilayer perceptron and gained an accuracy of more than 90%.

In this paper, we worked with gender, various age ranges, and dialects from different districts of Bangladesh. Since we have data diversity in our dataset, it helped us to achieve better accuracy than these two papers because with the more diversified data, it became easier for the computer to predict the result more accurately.

3. Proposed Method

3.1. Dataset Description

A dataset containing 1230 audio file (.wav format) was created for the experiment. Ten people from various parts of the country were asked to give their voice recordings for the experiment. Five of the participants were male, and five of the participants were female. They were from five different age groups. Table 1 gives an overview of the dataset created.

Table 1. Gender, age group and dialect used in the dataset.

Participant No.	Gender	Age Group	Dialect (Bangladesh)
1	Female	21-25	Munshiganj
2	Female	21-25	Dhaka
3	Female	21-25	Mymensingh
4	Female	01-05	Faridpur
5	Female	40-45	Kushtia
6	Male	16-20	Dhaka
7	Male	21-25	Kurigram
8	Male	21-25	Narayanganj
9	Male	21-25	Madaripur
10	Male	36-40	Faridpur

People say the same word differently based on their gender, age, and dialect. So, collecting voice recordings of people with different Bengali dialects, different genders, different age-groups helped this dataset to become diversified.

Table 2 gives an overview of the class labels of the dataset.

Table 2. Bengali Spoken Digit recorded

Bengali word	Bengali pronunciation	English word equivalent	English numerical equivalent
শূন্য	shun-no	zero	0
এক	e-k / a-k	one	1
দুই	du-i	two	2
তিন	ti-n	three	3
চার	cha-r	four	4
পাঁচ	pach/pan-ch	five	5
ছয়	ch-oy	six	6
সাত	sa-at	seven	7
আট	aa-at	eight	8
নয়	no-y	nine	9

3.2. Feature Extraction

MFCC or Mel Frequency Cepstrum Coefficient is a method of feature extraction from audio. Feature extraction is the process of discovering a value or a group of values that can be used to identify entities like objects, character, person, etc. MFCC is often used for speech recognition tasks [15]. MFCC features were extracted from the audio files using the "librosa" module of python [16] and appended into a python "numpy"- array [17]. Then the data were carefully labeled as 0,1,2,3,4,5,6,7,8 and 9.

3.3. Train Test Split

After feature extraction and labeling was completed, the dataset created was divided into "train" and "test" set. It was done using the test_train_split method from the "sklearn" module in python [18]. The dataset was divided by maintaining 80:20 ratio; 80% data was used for the training dataset and 20% data was used for the test dataset.

3.4. Feature Learning and Classification using CNN

In this experiment, Convolutional Neural Network (CNN) has been used for feature learning and classification purposes. Fig 1. describes the precise CNN architecture that was used for this experiment. It was created using the "Keras" module in python.

Train data was passed through the first 2D-convolutional layer (Conv2D) with a filter size of 32 and a relu activation function. Then the output of the first Conv2D layer was passed through the first BatchNormalization layer. This process is done two more times using the second Conv2D (filter size 48) -BatchNormalization and third Conv2D (filter size 120)- BatchNormalization pair layer. After that, the output is sent into a 2D-max pooling layer with a pool size of (2,2). After MaxPooling is completed, the output from that layer is passed down to the dropout layer with a 0.25 dropout rate. The dropout layer drops some values to reduce the chance of overfitting. This ends the Feature learning process.

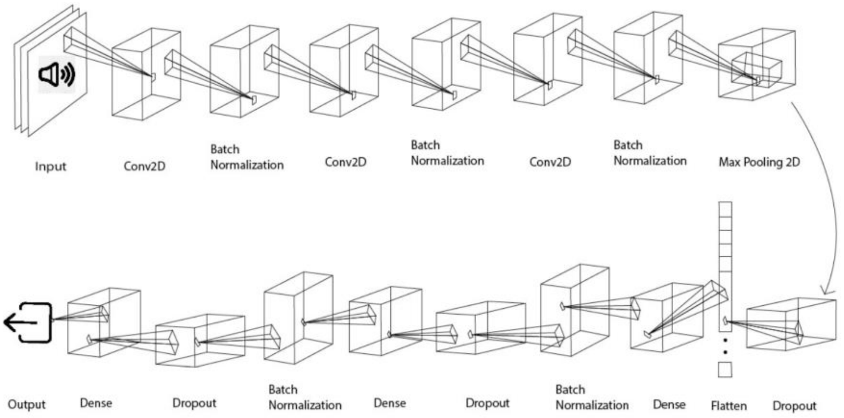

Fig 1: Architecture of the proposed approach using CNN

Then the resultant vector is transformed into a one-dimensional vector by using the Flatten layer. One-dimensional vector is easy to use for classification purposes, and thereby it was used to flatten the data. The 1D vector that came out of the Flatten layer is then passed through a series of layers consisting of a fully connected Dense layer with 128 units and relu activation function, BatchNormalization layer, Dropout layer with 0.25 dropout rate, another fully connected Dense layer with sixty-four units and relu activation, another BatchNormalization layer, another Dropout layer with a dropout rate of 0.4. Then the resultant vector is passed through a final fully connected Dense layer with ten units and softmax activation function. After that, we got the final output. The loss function and optimizer used in this model are categorical cross-entropy and adadelta.

4. **Result Analysis**

The experiment was run thrice, each time by taking a different size of the dataset. On the first run, 200 data were fed into the model. In the first run, an accuracy of 57.14% was achieved with 57% precision, 57% recall and 57% f1-score. In the second run, 500 data were fed into the model. In this run, an accuracy of 76% was found with 76% precision, 76% recall and 76% f1-score. In the final run, 1230 data was fed into the model. In this run, an accuracy of 98.37% was found with 98% precision, 98% recall and 98% f1-score. Table 3 describes the results found using the CNN model for different sizes of datasets.

Table 3. Results found by using the CNN model.

Data Size	Accuracy	Precision	Recall	F1- score
200	57.14%	57.00%	57.00%	57.00%
500	76.00%	76.00%	76.00%	76.00%
1230	**98.37%**	**98.00%**	**98.00%**	**98.00%**

Table 3 shows that the result drastically changes when a substantial amount of data is used. With an initial accuracy of 57.14% for 200 Data, the result jumps to an accuracy of 98.37% for 1230 Data. From the result, it can be safely presumed that higher accuracy is achievable if a more substantial amount of data is fed into the model.

Table 4. Results of each digit's precision, recall and f1-score found by using the CNN model.

Digit	Precision	Recall	F1-score
0	1.00	0.91	0.95
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	0.92	1.00	0.96
4	1.00	1.00	1.00
5	1.00	0.92	0.96
6	1.00	1.00	1.00
7	1.00	1.00	1.00
8	1.00	1.00	1.00
9	0.92	1.00	0.96

Table 4 shows precision, recall and f1-score of each digit that is found by using the CNN model for the dataset.
Fig 2. represents the improvement for different sizes of data graphically.

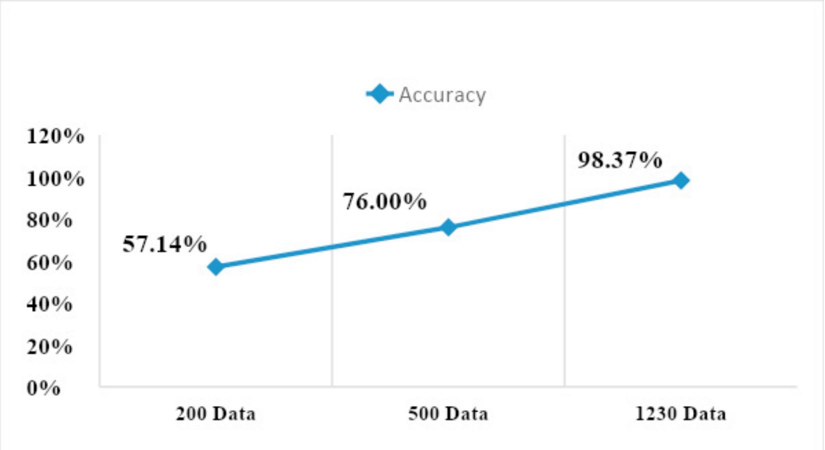

Fig 2: Improvement in result for different sizes of data fed into the CNN model

The results that were found using the CNN model is better than previous approaches. In paper [10], [11], [12] and [13] had different shortcomings. Paper [11], did not work with dialects, paper [13] didn't work with age-groups. However, in this paper, gender, dialects, age-groups, all these parameters were taken into account. The importance of taking these parameters into account can be seen in Table 5.

Table 5 shows the accuracy score of the used CNN model when only male and only female voice data was used for training and testing.

Table 5. Result found using data collected from different gender with CNN

Gender	Accuracy
Female	88.37%
Male	92.50%
Mixed	**98.37%**

Fig 3 depicts the impact of taking gender as a parameter created. The accuracy of the model has been increased after mixing both Male and Female dataset.

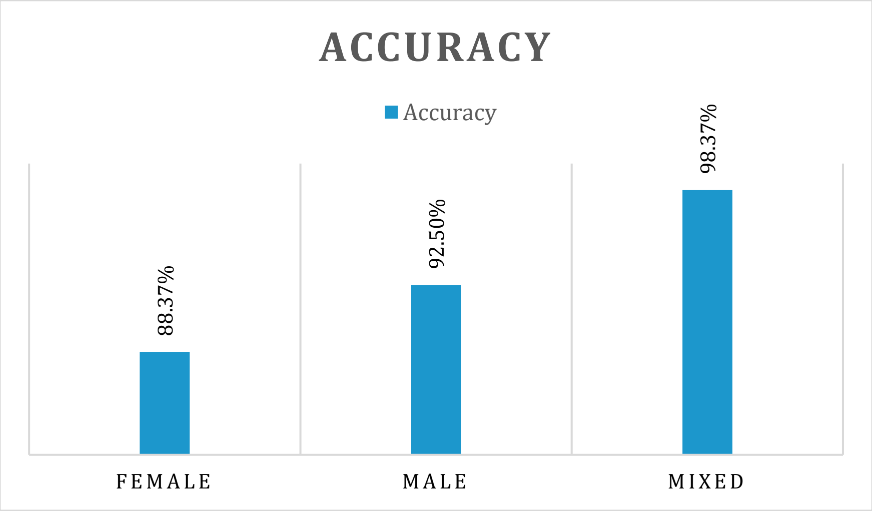

Fig 3: Improvement in result after mixing both male and female dataset

With this new approach with CNN for classifying Bengali spoken digits, the results also improved than the previous works that were done on this same topic. From Table 6, the comparison between results found by earlier approaches with the proposed method with CNN can be seen.

Table 6. Comparison between previous approaches and the proposed approach using CNN

Paper	Accuracy
Huque et al. [11]	98.05%
Gupta et al. [14]	More than 90%
Muhammad et al. [10]	Digit (0-5) more than 95% Digit (6-9) less than 90%
Proposed Approach	**98.37%**

Table 6 clearly depicts that the proposed approach using CNN gives better accuracy than the previous digit classification approaches on Bengali digits.

5. Future work

This paper tries to classify the Bengali spoken digit with high accuracy. Data was taken from 10 individuals with different gender, dialect, and different age-groups. But, all dialects of Bengali language have not been covered in this paper as collecting data from people with so many dialects can be a complicated job.
A bigger dataset covering all dialects of Bengali language from all age-groups can be created in the future. CNN proves to be suitable for classifying Bengali spoken digits, but other options can be tried as well for finding a better model.

6. Conclusion

Speech recognition is becoming more and more useful to humans due to technological improvement. Speech recognition in the Bengali language is also being used in many sectors. Bengali speech recognition, especially digits recognition can help in speech-based command for Internet-of-things devices. For example, an Intelligent traffic controlling system can get benefitted from this. Bengali spoken digit classification can also help in the field of human-computer interaction. Voice commands for cellular devices, Artificial Intelligence-based voice assistants can be created in Bengali. Bengali spoken digit classification can contribute a lot to all these rapidly growing fields.

In this paper, we propose a new approach towards classifying Bengali spoken digits with Convolutional Neural Network (CNN). The proposed model classifies spoken digit, spoken by ten different people with different gender, dialects, and of different age-groups with an accuracy of 98.37% which shows the viability of the proposed approach.

References

[1] What is speech recognition? - Definition from WhatIs.com. SearchCustomerExperience. https://searchcustomerexperience.techtarget.com/definition/speech-recognition (accessed September 30, 2019).
[2] Lane J. What Are The 10 Most Spoken Languages In The World?: Babbel. Babbel Magazine 2019. https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world (accessed September 30, 2019).
[3] Xiong W, Droppo J, Huang X, Seide F, Seltzer M, Stolcke A, et al. The Microsoft 2016 conversational speech recognition system. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2017. DOI:10.1109/icassp.2017.7953159.
[4] Graves A, Beringer N, Schmidhuber J. A comparison between spiking and differentiable recurrent neural networks on spoken digit recognition. Neural Networks and Computational Intelligence 2004.
[5] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 2005;18:602–10. DOI:10.1016/j.neunet.2005.06.042.
[6] Saxena B, Wahi C. Hindi Digits Recognition System on Speech Data Collected in Different Natural Noise Environments. Computer Science & Information Technology (CS & IT) 2015. DOI:10.5121/csit.2015.50303.
[7] S. R, Joseph A, K.k. AB. Isolated digit recognition for Malayalam- An application perspective. 2013 International Conference on Control Communication and Computing (ICCC) 2013. DOI:10.1109/iccc.2013.6731648.
[8] Dixit A, Vidwans A, Sharma P. Improved MFCC and LPC algorithm for bundelkhandi isolated digit speech recognition. Improved MFCC and LPC Algorithm for Bundelkhandi Isolated Digit Speech Recognition - IEEE Conference Publication 2016. https://ieeexplore.ieee.org/document/7755413 (accessed September 30, 2019).
[9] Ali Z, Abbas AW, Thasleema TM, Uddin B, Raaz T, Abid SAR. Database development and automatic speech recognition of isolated Pashto spoken digits using MFCC and K-NN. International Journal of Speech Technology 2015;18:271–5. DOI:10.1007/s10772-014-9267-z.
[10] Muhammad G, Alotaibi YA, Huda MN. Automatic speech recognition for Bangla digits. 2009 12th International Conference on Computers and Information Technology 2009. DOI:10.1109/iccit.2009.5407267.
[11] Huque S, Habib A, Babul M. Analysis of a Small Vocabulary Bangla Speech Database for Recognition. International Journal of Computer Applications 2016;133:22–8. DOI:10.5120/ijca2016907827.
[12] Sumon SA, Chowdhury J, Debnath S, Mohammed N, Momen S. Bangla Short Speech Commands Recognition Using Convolutional Neural Networks. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP) 2018. DOI:10.1109/icbslp.2018.8554395.
[13] Ghanty, S., Shaikh, S., Chaki, N., 2010. On Recognition of Spoken Bengali Numerals, International Conference On Computer Information Systems And Industrial Management Applications. IEEE, pp. 54-59.
[14] Gupta A, Sarkar K. Recognition of Spoken Bengali Numerals Using MLP, SVM, RF Based Models with PCA Based Feature Summarization. The International Arab Journal of Information Technology n.d.;15.

[15] Ganchev, T., Fakotakis, N., Kokkinakis, G., 2005. Comparative evaluation of various MFCC implementations on the speaker verification task, in: SPECOM. Proceedings of the SPECOM 1 (2005), Patras, Greece, pp. 191-194.

[16] librosa.feature.mfcc — librosa 0.7.0 documentation [WWW Document], 2019. [WWW Document]. Librosa.github.io. URL https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html (accessed 9. 30. 19).

[17] numpy.append — NumPy v1.17 Manual [WWW Document], 2019. [WWW Document]. Docs.scipy.org. URL https://docs.scipy.org/doc/numpy/reference/generated/numpy.append.html (accessed 9. 30. 19).

[18] sklearn.model_selection.train_test_split — scikit-learn 0.21.3 documentation [WWW Document], 2019. [WWW Document]. Scikit-learn.org. URL https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (accessed 9. 30. 19).