RESEARCH

Open Access

Amharic spoken digits recognition using convolutional neural network

Tewodros Alemu Ayall[1,4*], Changjun Zhou[1*], Huawen Liu[2], Getnet Mezgebu Brhanemeskel[3], Solomon Teferra Abate[3] and Michael Adjeisah[1]

*Correspondence:
ayalltewodros@zjnu.edu.cn;
zhouchagjun@zjnu.edu.cn

[1] School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
[2] Department of Computer Science, Shaoxing University, Shaoxing, China
[3] School of Information Science, Addis Ababa University, Addis Ababa, Ethiopia
[4] School of Natural and Computing Sciences, Interdisciplinary Centre for Data and AI, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom

Abstract

Spoken digits recognition (SDR) is a type of supervised automatic speech recognition, which is required in various human–machine interaction applications. It is utilized in phone-based services like dialing systems, certain bank operations, airline reservation systems, and price extraction. However, the design of SDR is a challenging task that requires the development of labeled audio data, the proper choice of feature extraction method, and the development of the best performing model. Even if several works have been done for various languages, such as English, Arabic, Urdu, etc., there is no developed Amharic spoken digits dataset (AmSDD) to build Amharic spoken digits recognition (AmSDR) model for the Amharic language, which is the official working language of the government of Ethiopia. Therefore, in this study, we developed a new AmSDD that contains 12,000 utterances of 0 (Zaero) to 9 (zet'enyi) digits which were recorded from 120 volunteer speakers of different age groups, genders, and dialects who repeated each digit ten times. Mel frequency cepstral coefficients (MFCCs) and Mel-Spectrogram feature extraction methods were used to extract trainable features from the speech signal. We conducted different experiments on the development of the AmSDR model using the AmSDD and classical supervised learning algorithms such as Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF) as the baseline. To further improve the performance recognition of AmSDR, we propose a three layers Convolutional Neural Network (CNN) architecture with Batch normalization. The results of our experiments show that the proposed CNN model outperforms the baseline algorithms and scores an accuracy of 99% and 98% using MFCCs and Mel-Spectrogram features, respectively.

Keywords: Automatic speech recognition, Spoken digit recognition, Amharic spoken digits recognition, Convolutional neural network, Speech feature extraction

Introduction

Speech is the most crucial part of communication between humans and machines. In the last decade, there has been a significant increase in the use of speech interfaces that enable hands-free human–machine communication. Speech interfaces make it possible for visually impaired people to communicate with machines straightforwardly. Speech instructions as machine input have various advantages because they are rapid, hands-free, and may be supplied remotely. Due to automating jobs necessitating ongoing interaction between humans and machines, automatic speech recognition (ASR) has received significant attention in recent decades [1, 2]. Spoken digits recognition (SDR) is a subset of supervised ASR in which the system can recognize each single digit. With the SDR, people can instruct machines via voice commands to perform various services such as dialing systems, airline reservation systems, certain bank operations, and price extraction. The SDR also simplifies the operation of technologies such as home automation and remotely controlled unmanned vehicles.

The goal of the SDR is to recognize human voice utterances from labeled audio data in the form of a signal. It uses various feature extraction techniques to encode features from signals and supervised machine learning (SML) [3] models to program intelligent machines without the involvement of a human. SML [3] is a type of machine learning that is driving forces in the modern computing era in speech recognition [4–7], and image classification [8–10]. Feature extraction is the process of keeping pertinent information from the speech signal while removing irrelevant and unwanted information [11]. Different features, such as Mel frequency cepstral coefficients (MFCCs) and Mel-Spectrogram features, can be extracted from wave signals. These extracted features are considered as input to SML models [12, 13]. The SDR is commonly designed using classical machine learning and deep learning approaches. Classical SML includes Hidden Markov models (HMMs), Gaussian mixture models (GMMs) HMMs (GMM-HMMs), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and Linear Discriminant Analysis (LDA). These classical models convert the input signal into the feature space of a specific problem using a simple structure. Therefore, they can not express complex functions when processing speech signals. Recently, the Deep learning (DL) model has been applied to ASR [5]. The DL model is well-known for building an artificial neural network (ANN) and capturing complex relationships between data features through multiple layers [14]. A Convolutional Neural Network (CNN) is an example of the DL model that preserves the inputs' spatial structure, and it was initially proposed for the recognition of handwritten digits [15]. The CNN model is commonly used for image, speech, video, text, and graph analysis [16]. Recently, it has shown more impressive recognition results in many languages of ASR [5, 17–19] than the conventional SML models.

There are approximately 7000 different languages spoken across the globe [20]. Amharic is the working language of the government of Ethiopia with a population of over 100 million people. Amharic is the most widely spoken language of Ethiopia and the second most commonly spoken Semitic language in the world after Arabic [21–23]. This language is spoken in different regions of Ethiopia especially in Addis Abeba, Gojjam, Gondar, Wollo, and North Showa with five different dialects [24]. Most people who speak this language are located in Ethiopia; however, there are speakers of Amharic in other countries, including Israel, Eritrea, Canada, the United States of America, Sweden, etc. The development of the ASR system for languages has impacted the creation of digital technologies and is also of significant economic value. Many researchers have investigated the SDR in various languages such as English, Arabic, Urdu, Hindi, Bangladesh, Uzbek, Pashto, Gujarati, etc. However, there has been little progress in the development of Amharic ASR [23, 25].

Researching and developing any language's ASR system requires a well-organized dataset. Preparing high-quality datasets is crucial for the success of designing the DL models. However, the lack of the dataset is a significant obstacle to develop machine learning models. Freely available dataset for the public is vital to develop any speech recognition systems, especially for under-resourced languages such as the Amharic language. However, there is no previously developed and publicly accessible Amharic spoken digits dataset (AmSDD) for Amharic spoken digit recognition (AmSDR) model. Therefore, we are motivated to develop this dataset and the recognition system. In speech recognition models, the different factors such as dialects [26] and genders [27] have an impact on the performance of the machine learning models. Therefore, we collected our dataset from volunteer speakers in different dialects, genders, and age distribution for the purpose of building the general machine learning model.

The contributions of this work are summarized as follows:

- We introduce a new AmSDD that contains a digit 0 (Zaero) to 9 (zet'enyi) from 120 volunteer speakers of different age groups, genders, and dialects with 10 repetitions of each digit. This dataset can be downloaded from here.[1]
- We propose AmSDR system using this AmSDD and various classical SML models to investigate the performance of the prediction and understanding of the nature of this dataset.
- To further improve the accuracy of the AmSDR, we also propose the DL model of CNN architecture with Batch Normalization and compare it with the baseline of classical SML models.
- We conducted extensive experimental evaluations to demonstrate the performance of the proposed work using MFCCs and Mel-Spectrogram feature extraction techniques.

The rest of the paper is structured as follows: "Related work" section presents related work, and "Amharic spoken digits recognition system" section elaborates on the steps to design the Amharic spoken digit recognition system. The experimental results and discussion are provided in "Experimental results and discussions" section. Finally, "Conclusion" section summarises the paper and gives directions for future works.

Related work

Many researchers have investigated SDR in various languages such as English, Arabic, Urdu, Hindi, Bangladesh, Uzbek, Pashto, Gujarati, etc. There are several works in English SDR (ESDR) [28–30]. Oruh et al. [28] presented ESDR using a deep forward ANN with hyperparameter optimization techniques, an ensemble method, RF, and regression. They used publicly available dataset [31], and Short-term Fourier transform with one hop encoding to extract features. Their deep forward ANN model scored 99.5% accuracy. Mahalingam and Rajakumar [29] presented ESDR using Long short-term memory (LSTM). The authors used the publicly available Free Spoken Digit Dataset (FSDD) [32], which contains 3000 utterances from six speakers with fifty repetitions for each digit.

[1] https://www.kaggle.com/datasets/tewodrosalemu/amharic-spoken-digits-dataset-amsdd.

They used wavelet scattering to extract features and got 97–98% accuracy with parameter tuning using Bayesian optimization. Nasr et al. [30] proposed ESDR using deep ANN architecture. The authors used FSDD and MFCCs to extract features and achieved 93% accuracy. Sarm et al. [33] proposed ESDR using ANN. They collected recordings from 30 male and 20 female speakers. The authors used Linear Prediction Coefficient feature extraction and Principal Component Analysis for variable reduction and achieved 82% accuracy. Taufik and Hanafiah [34] proposed an automated visual acuity test that can be performed on a standard computer with a microphone as an input device and a monitor. Visual acuity is assessed using a Snellen chart with digit Optotype and is based on the user's response in the form of spoken digits. The authors used MFCCs for feature extraction and the CNN model. Their model achieved 91.4% accuracy.

Numerous researchers have investigated Arabic SDR (ASDR). Wazir et al. [35] proposed ASDR using LSTM and collected 1040 audio samples, and divided it into 840 for training and 200 for testing. They used MFCCs for feature extraction and achieved 69% accuracy. Zerari et al. [36] presented a comprehensive framework for ASDR and spoken TV commands via LSTM and ANN. The authors used both MFCCs (dynamic and static features) extraction strategies as well as the Filter Banks coefficient. LSTM or Gated Recurrent Unit (GRU) architecture is utilized for encoding the sequences and is introduced to a Multi Layer Perceptron network (MLP) for recognition. Their model reaches 96% accuracy. Azim et al. [18] proposed ASDR using CNN model. They used 8800 utterances to represent all digits with ten repetitions among 88 speakers. The authors utilize MFCCs for feature extraction, and their CNN model scored 99% accuracy.

Urdu SDR (USDR) has been proposed in [19, 37, 38]. Hasnain and Awan [37] investigated the frequency analysis of USDR via Fast Fourier Transform (FFT) feature extraction. The authors experimented on 15 speakers and observed a strong correlation between numerous speakers' frequency contents of the same word. Ali et al. [38] proposed USDR using RF, SVM, and LDA. The experiment was conducted on ten speakers and MFCCs were used for feature extraction. They got 73% accuracy on SVM, which is better than RF and LDA. Aiman et al. [19] proposed CNN model for USDR. They collected 25,518 audio samples from 740 participants. The authors extracted Mel-Spectrogram from the audio signal and made the classification of digits using different algorithms. Their proposed CNN model reaches 97% accuracy.

Several works have been proposed in Bangali SDR (BSDR). Gupta and Sarkar [39] proposed BSDR in noisy and noise-free environments by multiple speakers with different dialects. MFCCs and Principal Component Analysis were used for feature extraction and feature reduction. The authors designed using MLP, RF, and SVM, scored more than 90% accuracy. Paul et al. [40] proposed BSDR using GMMs and MFCCs for feature extraction and achieved 91.7% prediction accuracy. Riffat Sharmin et al. [17] proposed BSDR using CNN model. They used MFCCs for feature extraction and achieved 98.37% accuracy. Das et al. [41] proposed the mixed Bangla-English SDR Using CNN model. They used the combination of Bangla-English datasets and MFCCs feature extraction and achieved 87% accuracy.

In other languages, the SDR also has been investigated by many researchers. Dhandhania et al. [42] proposed Hindi SDR using the HMMs. They collected 1000 utterances from 20 speakers, used MFCCs for feature extraction, and achieved 75% accuracy. Zada

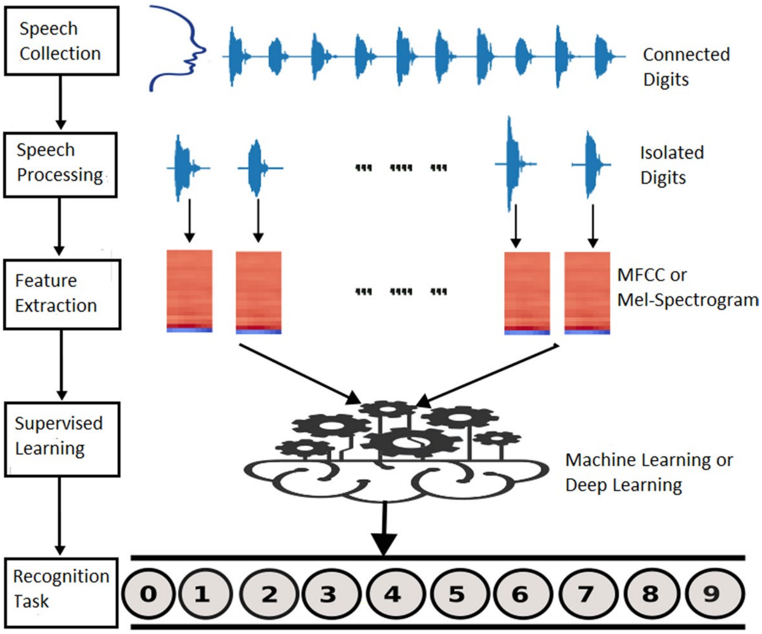


Fig. 1 The design of SDR system

and Ullah [43] proposed Pashto SDR via CNN model. They used 500 utterances from 25 male and 25 female speakers with ten repetitions for each digit, MFCCs for feature extraction, and achieved 84.17% accuracy. Musaev et al. [44] proposed Uzbek SDR via CNN model. They collected 19 women speakers with 10 repetitions for each digit and used spectrogram for feature extraction. Their model scored 99.7% accuracy. Renjith et al. [45] proposed Malayalam SDR via HMMs. They used MFCCs for feature extraction and attained 87% accuracy. Dalsaniya et al. [46] proposed Gujarat SDR via naive ANN classifier. The authors collected audio samples from 20 speakers in different regions, genders, and age groups, with ten repetitions for each digit. MFCCs were used to extract features, and 75% accuracy was achieved.

Amharic spoken digits recognition system

Before performing a recognition task, there are basic procedures to follow. In this work, we followed five steps: speech collection, speech preprocessing, feature extraction, choosing supervised learning models, and applying recognition tasks. Figure 1 shows the detailed procedure of the design of the SDR system. Speech collection is performed by selecting the target speaker group representing the whole population. The collected speech is connected digits; thus, a preprocessing step is needed. In the preprocessing step, we made a segmentation to change connected digits to isolated digit audio samples. We chose parameters for the audio samples such as sample rate, format type, file renaming, etc. Performing a prediction on an audio signal is time-consuming; therefore, selecting the appropriate feature extraction method is essential to compute the recognition tasks efficiently and to get a remarkable prediction result. Depending on the machine learning model, features are required to be reshaped. Due to SDR being a supervised task, well-known supervised algorithms are used to investigate the performance of the prediction. The recognition performance of these well-known algorithms has not

Table 1 Amharic digits script and pronunciation

Digits	Amharic digits script	Pronunciation
0	ዜሮ	zaero
1	አንድ	ānidi
2	ሁለት	huleti
3	ሦስት	sositi
4	አራት	ārati
5	አምስት	āmisiti
6	ስድስት	sedisiti
7	ሰባት	sebati
8	ስምንት	siminiti
9	ዘጠኝ	zet'enyi

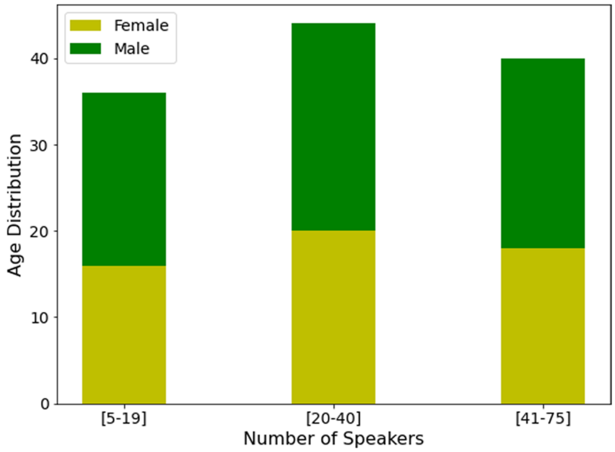


Fig. 2 Number of speakers age distribution

reached a satisfactory level due to the characteristics of the languages and the model itself. Therefore, we propose a DL framework.

Speech collection

We prepared AmSDD for our digit recognition task. The primary reason for our motivation is to develop a general Amharic speech dataset to make automatic AmSDR efficient and robust. To the best of our knowledge, there is no publicly available AmSDD. Therefore, this is a new publicly available AmSDD dataset that can be used by other researchers to design the DL model.

The Amharic language has its own scripts and pronunciations as described in Table 1. We collected Amharic speech from volunteer speakers of various ages, genders, and dialect groups. There were 120 participants in three age groups: 5–19, 20–40, 41–75, and in five dialects such as Addis Abeba, Gojjam, Gondar, Wollo, and North Showa. Figure 2 shows participants' age distribution, and the majority of participants ranged in age from 19 to 40 years old. Male participants slightly outnumbered female participants, as shown in Fig. 2. We recorded each audio sample using a mobile recorder with a sample rate of 44.1 kHz in mp4 format and different environments, including normal, noisy, and closed

Table 2 Characteristics of AmSDD

Attributes	Values
Sampling rate	16 kHz
Number of quantization (bits)	16 bit
Number of channel	mono
Audio file format	.wav
Number of speakers according to genders	Male and female
Age distribution	Children, young and middle age
Recording environment	Normal life, closed room and with noise
Duration	Less than 1 s
Dialects	Addis Ababa, Gojjam, Gondar, Wollo and North Showa
Number of speakers	120
Number of tokens per speaker	100
Number of digits	10
Number of repetitions per digit	10
Total number of utterances	12,000

room, to make the dataset more diverse and challenging for prediction. A total of 12,000 utterances were recorded. Each class has an equal number of samples, which is 1200 utterances.

Speech preprocessing

The initial recorded audio samples are continuous speech with a high sample rate. Therefore, speech preprocessing is needed to make isolated digits and to downsample its sample rate. We performed manual and automatic segmentation techniques to create isolated digits. Before applying segmentation, all audio samples were changed to 16 kHz sample rate, a mono channel, 16-bit float datatype, and wav file format. However, manual segmentation requires more labor-intensive work. Therefore, to reduce the preprocessing time, first we manually segmented a single digit continuous spoken with 10 repetitions, and then each ten continuous spoken digits is again segmented using automatic segmentation. We used a python Pydub [47] package for automatic segmentation which is a general purpose audio processing functionality. From this package, we used *split_on_silence* method which returns splitting audio segment on silent sections. We provided these two parameters *min_silence_len* is 250 and *silence_thresh* is −60 to this method. Naming convention of each file was <SpeakerID>_<Digit>_<Repetition>. For example, in an audio file named S1_01_Five_10.wav, S1 indicates a speaker 1, Five represents a digit 5, and 10 represents how many times the speaker repeated the digit 5. All audio samples are arranged based on their class type. Table 2 describes the detailed characteristics of AmSDD.

Feature extraction

The feature extraction method is the most crucial component in the design of the ASR system. It assists the system in identifying the speaker by extracting relevant features from the input signal [11]. Although it is theoretically possible to recognize speech directly from a digitized waveform, extracting some features is preferable to minimize

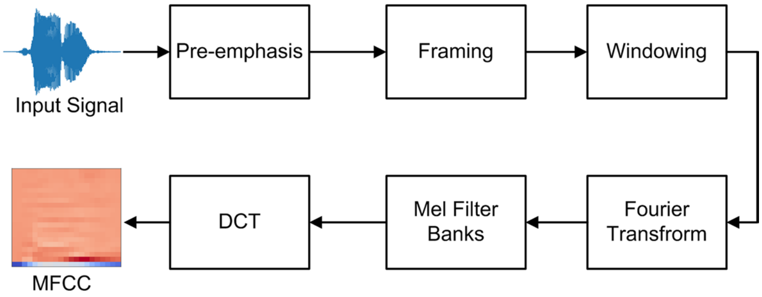


Fig. 3 MFCCs feature extraction

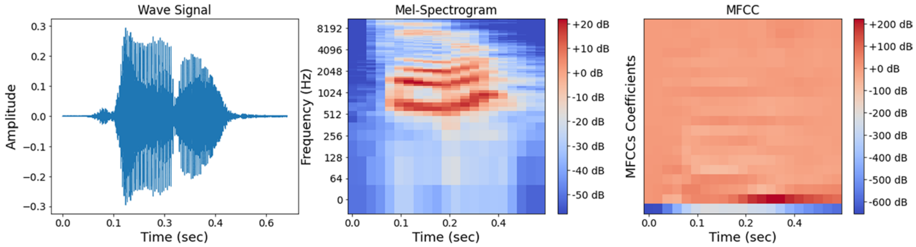


Fig. 4 Class 0: ዜሮ

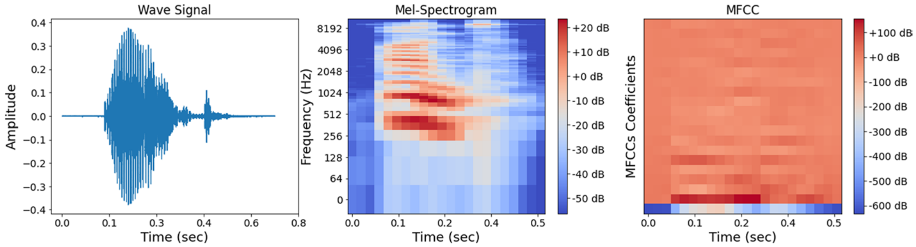


Fig. 5 Class 1: አንድ

variability [48] due to the high variability of speech signals. There are different types of feature extraction methods [49]. However, in this study, we used the most popular feature extraction method for SDR, such as Mel-Spectrogram and MFCCs.

Mel-spectrogram

A spectrogram is a graphical representation of the frequencies of a given signal as they change over time. One axis represents time, the second axis represents frequencies, and the colors represent the magnitude (amplitude) of the observed frequency at a given time in a spectrogram representation. Strong frequencies are represented by bright colors. Smaller frequencies (0–1 kHz) are particularly powerful. The audio signal is divided into equal-length segments (frames) to create a spectrogram. The STFT is then computed for each frame. The logarithmic Mel-Scaled filter bank is applied to the Fourier transformed frames to generate the Mel-Spectrogram [50]. The Mel-Spectrogram feature for each class zero to nine are illustrated in Figs. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.

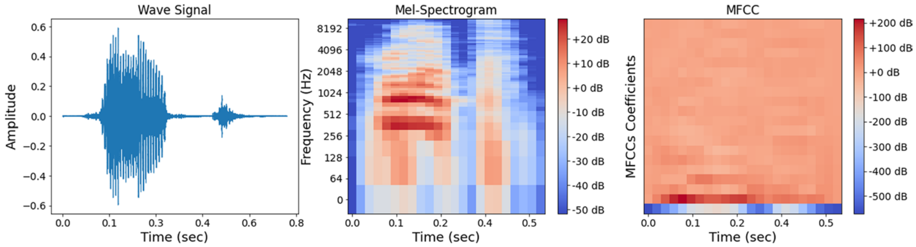


Fig. 6 Class 2: ሁለት

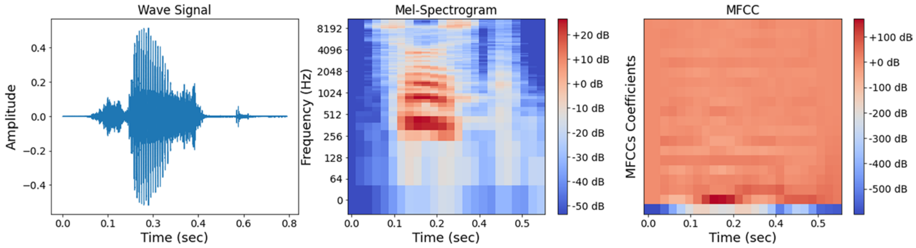


Fig. 7 Class 3: ሦስት

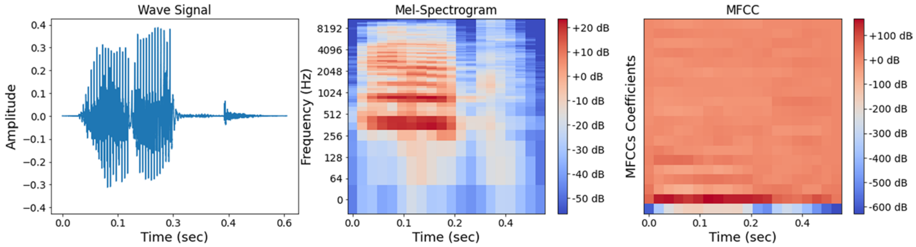


Fig. 8 Class 4: አራት

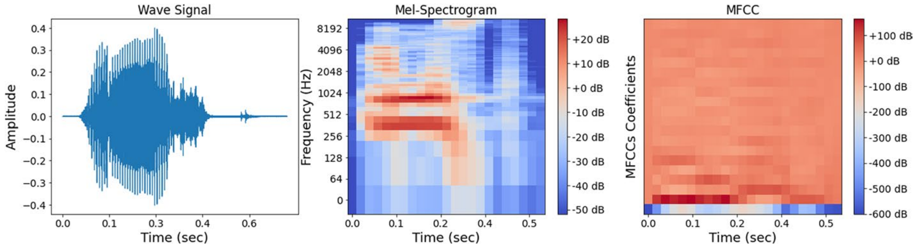


Fig. 9 Class 5: አምስት

Mel-frequency cepstral coefficients

MFCCs are the most dominant feature extraction method for SDR [11, 49]. The cepstral representation of an audio clip is used to generate MFCCs. The block diagram in Fig. 3 shows the steps involved while computing MFCCs. In the process of MFCCs feature extraction, first, the analog continuous time varying input signal is given as an input. Since high frequencies in the input speech signal often have a smaller magnitude than

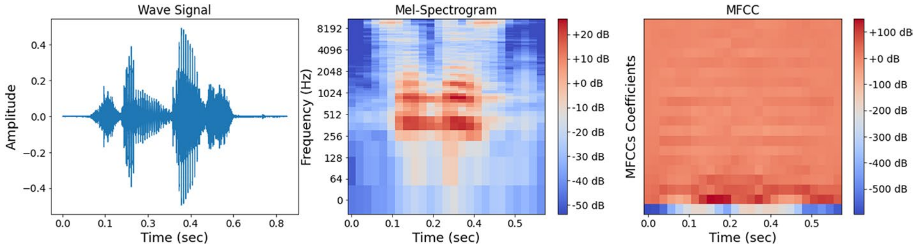


Fig. 10 Class 6: ስድስት

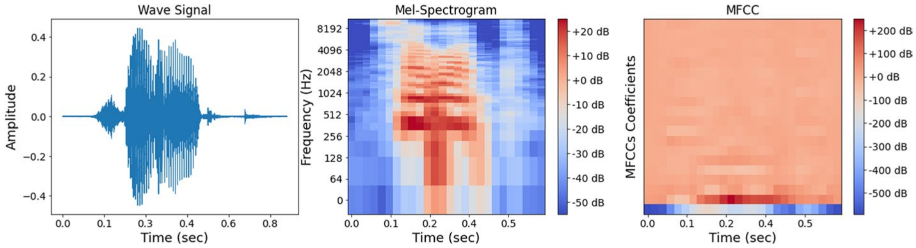


Fig. 11 Class 7:ሰባት

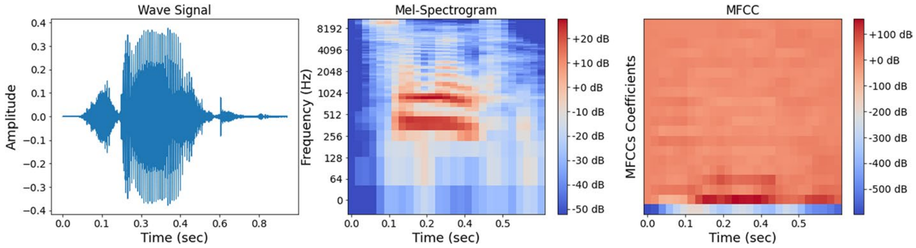


Fig. 12 Class 8: ስምንት

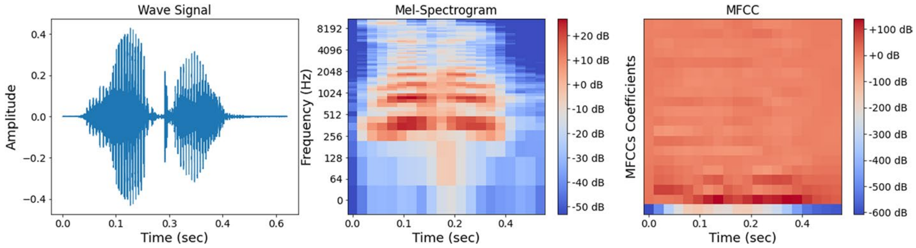


Fig. 13 Class 9: ዘጠኝ

lower frequencies, a pre-emphasis filter is applied to amplify the high frequencies. After pre-emphasis, the signal is split into short-time frames. This step is necessary because frequencies in a signal change over time. This can be extracted from each input speech signal with a frame size of 25 ms, which is considered as stationary segment. After framing, the next step is windowing on each segmented frame to minimize spectral distortion of the signal. This is done using the windowing function like hamming window. The next step is to convert each frame using a fast Fourier transform (FFT), which is also called Short Time Fourier Transform (STFT). A Fourier transform converts the time to frequency and vice versa. The final step in the MFCC is Filter Bank analysis, which is computed for each frame by applying the discrete cosine transform (DCT). The DCT of the log power spectrum on a nonlinear mel scale represents the short-time power spectrum of an audio clip. DCT is applied because the output of the filter bank is highly correlated, which will become difficult for the machine learning algorithm to deal with it [51, 52].

Visualization

We demonstrated each class's audio sample in wave signal, Mel-Spectrogram, and MFCCs as illustrated in Figs. 4, 5, 6, 7, 8, 9, 10, 11, 12, 13. To visualize audio samples of each class, we loaded audio samples using the Librosa library [53]. From this visualization, we can understand that any representation of the waveform, Mel-Spectrogram, and MFCCs for each class digit is unique.

Supervised machine learning

SML makes use of data that has been labeled. The data is labeled because it consists of pairs of inputs that a vector can represent and their corresponding desired output. The vector can be used to represent the input, and the desired output can be described as a supervisory signal. Because the correct output is already known, the learning mechanism is said to be supervised. Suppose that we have a training set $\left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{m}$ draw from a joint distribution $p(x, y)$, $x \in X$, $y \in Y$, where X is MFCCs or Mel-Spectrogram features, Y is a labels and m is the number of training sample. The goal of supervised learning is to get a decision function $f : X \rightarrow Y$ that correctly predicts the output of unseen input from the same distribution. This prediction is called a supervised automatic SDR task. This problem can be solved using supervised learning models such as LDA, KNN, RF, SVM, and CNN.

Linear discriminant analysis

LDA [54] is the SML and dimensionality reduction that can be used to classify data as well as reduce the number of dimensions of the data. To achieve a higher level of separability, the LDA first transforms the data into a matrix and then calculates inter-class and intra-class variance. Second, the distance between the mean and the samples of each class is computed across all test cases. Finally, it builds the lower dimensional space while minimizing the intra-class variance and maximizing inter-class variance. LDA is also used in SDR task [38].

K-nearest neighbors

KNN is a simple yet effective SML that is used in a wide range of applications. Given a test audio to be classified based on feature extraction type, the algorithm searches for the k nearest neighbors among the pre-classified training audio sample using some similarity measure, ranks those k neighbors based on their similarity scores, and uses the categories of the k nearest neighbors to predict the category of the test audio using the ranked scores of each as the prediction weight.

Support vector machine

SVM, is a well-known example of SML that makes use of a hyperplane to divide the training data in order to categorize future predictions. The hyperplanes serve as decision boundaries that assist in the categorization of the data points. They are used to divide a dataset into two different classes. The goal of SVM is to create a dividing hyperplane that is maximally distant from both classes. This helps to organize the data in accordance with the category to which it belongs. It accomplishes this by locating the support vectors that have the greatest possible margin space between them. SVM is applied in many classification tasks in various domains and is also applicable for SDR task [38, 39].

Random forest

RF [55] is an example of SML based on an ensemble classifier that combines the predictions of many decision trees through the use of majority voting in order to output the class for a given input vector. Each decision tree that is a part of the ensemble process to select a subset of features at random in order to determine which split is the most optimal at each node of the decision tree. During the process of training the model, each tree is presented with a random selection of the data. This may cause some trees to use the same data more than once. The purpose of this is to reduce the variance of the model, which in turn reduces the disparity in the scores that were predicted based on the results. When dividing up the nodes in the trees, only a small subset of the features should be used. This is done to prevent the model from overfitting, which occurs when the model uses the training data to inflate the predictions made by the model. The process of using the average of the predictions made by each tree to determine the overall category of the data is referred to as bootstrap aggregating. This method is used when making predictions using RF.

Convolutional neural network

CNN is a type of ANN that helps to design the DL model. Even though CNN has made significant progress in image recognition and ASR, it has not been applied to AmSDR. In this work, we propose CNN for AmSDR, which consists of a number of layers such as a convolutional layer, max pooling layer, dropout layer, flatten layer, fully connected layer, and softmax layer to achieve high recognition performance.

Let $\mathbf{X}$ be a sequence of an acoustic feature that $\mathbf{X} \in \mathbb{R}^{C*F*T}$ where C is a number of channels, F is a number of frequency bands, and T is a time length. The convolutional layer multiplies the input $\mathbf{X}$ with a set of kernel filters. We have used three convolutional layers, as shown in Fig. 14. We used an activation function to normalize the input and produce an output, which is then passed forward to the next layer. The activation function introduces nonlinearity into the output, allowing neural networks to solve nonlinear problems. Sigmoid, Than, ReLU, and LReLU are examples of activation functions [56]. ReLU (α) is a widely used activation function in convolutional networks. Let x be an input, and a function $\alpha(x) = max(x, 0)$; if the value of x is negative, $\alpha(x)$ will be zero;

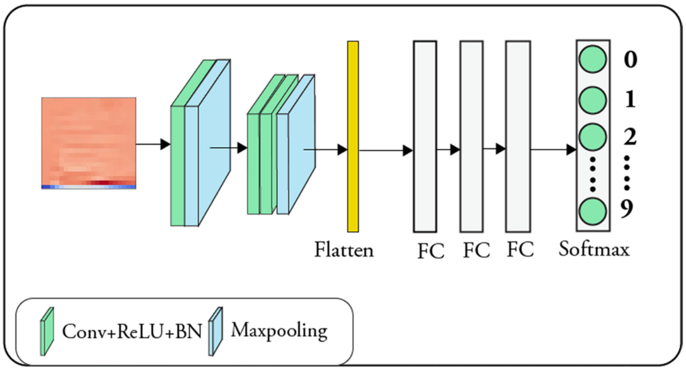


Fig. 14 Proposed CNN architecture

otherwise, the value of $\alpha(x)$ will be equal to x. Therefore, we used the ReLU activation function in each convolutional layer.

After obtaining the feature maps, a pooling (sub-sampling) layer is added alongside the convolutional layers. The pooling layer's task is to reduce the spatial size of the convolved feature and training time while preventing overfitting. There are two types of pooling: maximum (max) and average pooling. In this work, we used max pooling. In max pooling, the maximum value is chosen from a given kernel size and located in the output matrix [57]. Batch Normalization (BN) is a widely used technique for training deep neural networks faster and more consistently. Therefore, we used BN in each convolutional layer.

The output of the convolutional and pooling operations was a two-dimensional matrix. Therefore, this matrix must be flattened before being fed to the fully connected layer (FC). Therefore, the FC layers are added to CNN architectures at the end, and they are the ones that are responsible for carrying out the classification process. After flattening layers, we used three FC layers, and each input is connected to all the neurons. This layer operates on a one-dimensional input tensor.

Experimental results and discussions

Experimental setups and configuration of parameters

The experiment was conducted on Ubuntu operating system 22.04, the 11th Gen Intel®Core™i9-11950 H CPU @2.60 GHz 2.60 × 16, 32.00 GB RAM Dell computer and NVIDIA T600 laptop GPU with 4 GB video memory card. Manually and automatically segmented speeches were prepared using Audacity software [58] and Python's pydub [47] package, respectively. The proposed CNN model implementation and feature extractions were performed using Pytorch [59]. We used Adam optimizer, learning rate of 0.0001, 64 batch size, 100 epochs and ReLU activation function for the training of the CNN model. To implement LDA, KNN, SVM, and RF, we used Scikit-learn [60]. We extracted MFCCs and Mel-Spectrogram features based on the parameters described in Table 3. For MFCCs, we used all parameters in Table 3. However, for Mel-Spectrogram, we used all parameters in Table 3 except the number of Mel bands and Cepstral coefficients. We used 128 number of Mel Filter Banks for the Mel-Spectrogram feature, but Cepstral coefficients are not required it. Since the wave signal of each utterance has a different size, we used padding for the shorter

Table 3 Feature extraction parameters

Parameters	Values
Sampling rate	16 kHz, 16 bit
Fast Fourier transform	512
Hop length	256
Applied window	Hamming
Number of Mel Filter Banks	23
Cepstral coefficients	13

Table 4 The detailed proposed CNN model parameters using MFCCs feature

Types of layer	Dimension	Remarks
Input	(1, 13, 63)	MFCCs
Conv2d	(32, 12, 62)	kernel 2 × 2, stride = 1, ReLu activation
Maxpool2d	(32, 6, 31)	Max pool 2 × 2
BatchNorm2d	(32, 6, 31)	N.A
Conv2d	(64, 5, 30)	Kernel 2 × 2, stride = 1, ReLu activation
BatchNorm2d	(64, 5, 30)	N.A
Conv2d	(128, 4, 29)	Kernel 2 × 2, stride = 1, ReLu activation
Maxpool2d	(128, 2, 14)	Max pool 2 × 2
BatchNorm2d	(128, 2, 14)	N.A
Dropout	(128, 2, 14)	Dropout rate = 0.4
Flatten	3584	N.A
Linear	256	ReLu activation
Dropout	256	Dropout rate = 0.4
Linear	128	ReLu activation
Dropout	128	Dropout rate = 0.4
Linear	10	Softmax activation

wave signal to make it equal to the longer signal. The dimension of the extracted MFCCs feature is (X, n, m), where X is the number of training or test sample, n is Cepstral coefficients and m is a number of time frames (the sample rate times the duration of audio divided by hop length). This dimension depends on the extracted MFCCs features and the length of the signal. In our case, the dimension of MFCCs is (X, 13, 63). Similarly, the dimension of the Mel-spectrogram feature is (X, 128, 63), where 128 is a number of Mel Filter Banks. We investigated the recognition accuracy using well-known supervised learning algorithms such as LDA, KNN, SVM, and RF. To feed the MFCCs and Mel-spectrogram as input to LDA, KNN, SVM, and RF, the shape of the input should be changed to a one-dimensional feature vector. Therefore, the length of the MFCCs and Mel-Spectrogram feature vectors are $13 * 63 = 819$ and $128 * 63 = 8064$, respectively. Finally, we used (X, 819) and (X, 8064) feature vectors for MFCCs and Mel-Spectrogram, respectively, to train or test the above algorithms. We observed that the accuracy of these models is not satisfactory; thus, we designed the deep CNN as shown in Fig. 14. For the proposed CNN model, the dimensions of MFCCs and Mel-Spectrogram features are (1, 13, 63) and (1, 128, 63), where 1 is a mono channel, respectively. We developed the CNN with different layers as depicted in Table 4.

Performance evaluation metrics

We used the following performance evaluation metrics: accuracy, precision, recall, and F1-Score. Because the SDR is a multi-label classification with a class imbalance problem, test accuracy is not the ideal metric for evaluating the model. Thus, the classification report is more appropriate to display on a class-by-class basis. True Positive (TP) and True Negative (TN) represent the number of positive and negative samples identified correctly, respectively. On the other hand, False Positives (FP) and False Negatives (FN) represent the number of positive and negative samples identified incorrectly. Equations (1–4) describe the mathematical aspect of the metrics [61].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100, \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \times 100, \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \times 100, \tag{3}$$

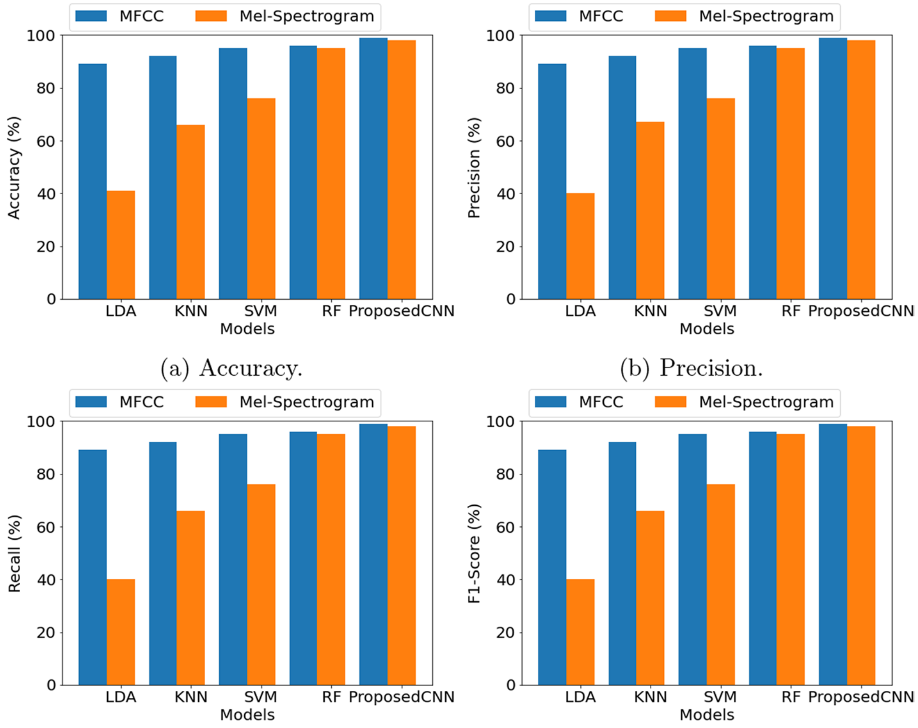
$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \times 100. \tag{4}$$

Experimental results

We split the AmSDD into 80% and 20% for training and testing cases, respectively. Each model was trained five times with a different random train/test split, and the average test data results were presented. We investigated the performance of the AmSDR using MFCCs and Mel-Spectrogram features, as shown in Fig. 15. The recognition performance of LDA, KNN, and SVM using the MFCCs feature is far better than the Mel-Spectrogram feature. However, MFCCs and Mel-Spectrogram feature extraction for RF and proposed CNN showed almost near recognition results, as shown in Fig. 15. Our proposed CNN model using MFCCs scored accuracy, precision, recall, and F1-Score for 99%, 99%, 99.01%, 99%, respectively. Due to the MFCCs feature showing better results than Mel-Spectrogram, we used the MFCCs feature to compare with other models and further analyze the proposed CNN model.

Based on MFCCs features, the proposed CNN model outperformed LDA, KNN, SVM, and RF by absolute accuracy margins of 10.10%, 7.07%, 4.04%, 3.03%, respectively, as shown in Fig. 15a. The proposed CNN outperformed LDA, KNN, SVM, and RF by precision margins of 10.10%, 7.07%, 4.04%, and 3.03%, respectively, as illustrated in Fig. 15b. The proposed CNN model outperformed in terms of recall by 10.10% of LAD, 7.07% of KNN, 4.04% of SVM, and 3.03% of RF, as shown in Fig. 15c. Similarly, it outperformed LDA, KNN, SVM, and RF by F1-Score margins of 10.10%, 7.07%, 4.04%, 3.03%, respectively, as depicted in Fig. 15d.

The confusion matrices of our model using the MFCCs features are shown in Fig. 16. The diagonal values illustrate the true class 1 predicted as class 1, and the same is true



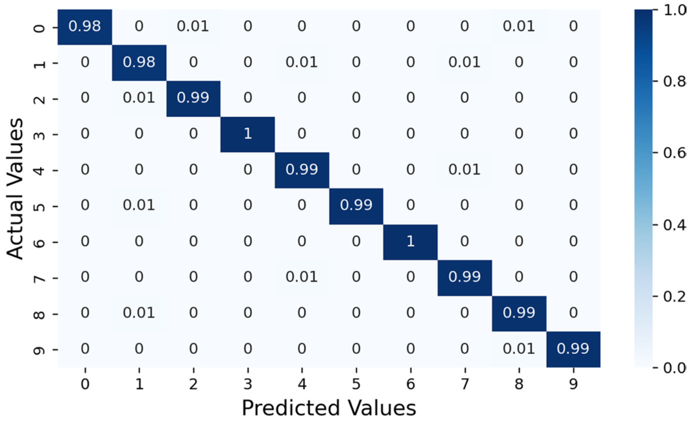
(a) Accuracy.

(b) Precision.

(c) Recall.

(d) F1-Score.

Fig. 15 Recognition performance of the models



	0	1	2	3	4	5	6	7	8	9
0	0.98	0	0.01	0	0	0	0	0	0.01	0
1	0	0.98	0	0	0.01	0	0	0.01	0	0
2	0	0.01	0.99	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	0.99	0	0	0.01	0	0
5	0	0.01	0	0	0	0.99	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0.01	0	0	0.99	0	0
8	0	0.01	0	0	0	0	0	0	0.99	0
9	0	0	0	0	0	0	0	0	0.01	0.99

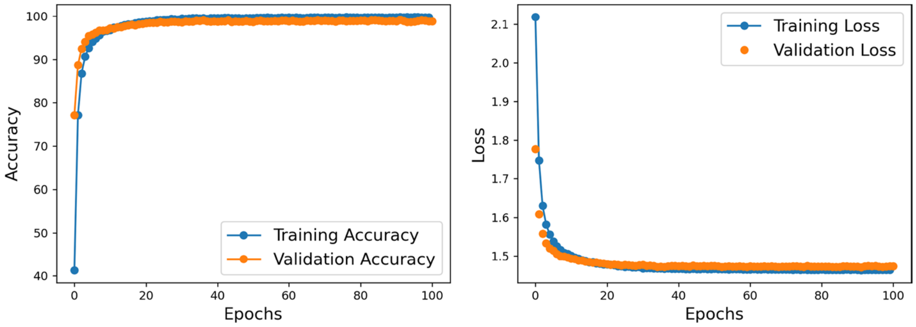
Fig. 16 Confusion matrix

for other digit classes. We observed from this confusion matrix which classes were wrongly predicted. For example, class 0 and 1 are 98% correctly predicted, and the other 2% is wrongly classified as other classes. The model wrongly predicted class 0 as 1% to class 2 and 1% to class 8. Similarly, the model wrongly predicted class 1 as 1% to class 4 and 1% to class 7. This class 0 and 1 prediction result affects the model's overall accuracy. In general, our confusion matrix shows good prediction results. Further, to analyze the proposed CNN, we calculated the individual class level accuracy, precision, recall, and F1-Score from this confusion matrix as shown in Table 5. These

Table 5 Performance evaluation per class level

Classes	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
0	98.00	100.00	98.98	98.00
1	98.80	97.00	97.51	98.80
2	99.00	99.00	99.00	99.00
3	100.00	100.00	100.00	100.00
4	99.00	98.00	98.50	99.00
5	99.00	100.00	99.49	99.00
6	100.00	100.00	100.00	100.00
7	99.00	98.00	98.50	99.00
8	99.00	98.00	98.50	99.00
9	99.00	100.00	99.49	99.00
Mean	**99.00**	**99.01**	**99.00**	**99.00**

Bolded value indicate the good experimental results



(a) Accuracy curve.

(b) Loss curve.

Fig. 17 Learning convergency curve for accuracy and loss

Table 6 The performance of our proposed CNN in other languages

Languages	No. of utterances	Training size (%)	Test size (%)	Feature extraction	Accuracy (%)
English [32]	3000	80	20	MFCCs	98.33
				Mel-spectrogram	97.50
Gujarati [46]	1940	80	20	MFCCs	96.80
				Mel-spectrogram	88.00

results indicated that our model showed good performance scores at the class level and overall class.

We plotted the learning curve for accuracy and loss for training and validation as shown in Fig. 17. Fig. 17a, b show the accuracy and loss curves of training and validation, respectively. This learning curve is used to detect the overfitting and underfitting problems of the model. Therefore, we observed that our model ideally learns with training and validation samples without underfitting or overfitting problems.

As described in Table 6, we checked the performance of our proposed CNN model with other two open spoken digits datasets such as English and Gujarati. Therefore, our model showed comparable accuracy. We investigated the performance of our AmSDR

Table 7 Latest results in other languages SDR

Languages	Models	Feature extractions	Accuracy (%)
English [28]	DFNN	MFCCs	99.50
Arabic [18]	CNN	MFCCs	99.00
Urdu [19]	CNN	Mel-Spectrogram	97.00
Bangali [17]	CNN	MFCCs	98.37
Hindi [62]	Pattern network	MFCCs	96.80
Gujarati [63]	CNN	MFCCs	98.70
Portugese [64]	SVM	Line spectral frequencies (LSF)	99.33
Pashato [65]	SVM	Prosodic	91.50
Amharic (ours)	CNN	MFCCs	99.00

Table 8 The effect of genders in recognition accuracy

Training type	Training size (%)	Test type	Test size (%)	Feature extraction	Accuracy (%)
Females	42.5	Males	57.5	MFCCs	81.50
				Mel-spectrogram	73.20
Males	57.5	Females	42.5	MFCCs	92.50
				Mel-spectrogram	82.50
Both	42.5	Both	57.5	MFCCs	97.62
				Mel-spectrogram	96.54
Both	57.5	Both	42.5	MFCCs	98.50
				Mel-spectrogram	97.64

model with the state-of-the-art other language SDR models based on the attributes described in Table 7. Table 7 shows nine other languages' models, feature extraction, and accuracy. This result indicated that our methodology and method are practical approaches. Therefore, a more attractive result was found in our Amharic language SDR as described in Table 7.

Impact of various factors on model performance

We showed the impact of genders, dialects, sample rate, number of MFCCs, learning rate, and batch size in our model. We showed the effects of gender on AmSDR as described in Table 8. In our dataset, we recorded 42.5% of female and 57.5% of male speakers. To check the effect of gender, we trained the model only on females' speech and tested it on males' speech and vice versa. Therefore, we observed that the model performance is greatly reduced by training and testing using different genders. To confirm this effect on the model may be in data splitting, we randomly split the dataset by 42.5% and 57.5% for training and testing including both speeches and vice versa. In both cases, we observed that the accuracy of the model is increased. Therefore, we conclude that training a model using only female or male speeches can not guarantee the performance of the model.

We showed the effect of dialects in our model as described in Table 9. Out of the five dialects, we trained the model by combining four dialects and tested it with the remaining dialect. For example, we trained our model using Addis Ababa, Gondar, Gojjam,

Table 9 The effect of dialects in recognition accuracy

Training type	Training size (%)	Test type	Test size (%)	Feature extraction	Accuracy (%)
Addis Ababa + Gondar + Gojjam + North Shewa	84.99	Wollo	15.01	MFCCs	91.00
				Mel-spectrogram	87.28
Addis Ababa + Gondar + Gojjam + North Wollo	81.67	North Shewa	18.33	MFCCs	94.50
				Mel-spectrogram	92.40
Addis Ababa + Gondar + North Shewa + Wollo	80.00	Gojjam	20.00	MFCCs	93.00
				Mel-spectrogram	87.80
Addis Ababa + Gojjam + North Shewa + Wollo	74.18	Gondar	25.82	MFCCs	91.50
				Mel-spectrogram	85.00
Gondar + Gojjam + North Shewa + Wollo	75.83	Addis Ababa	24.17	MFCCs	92.50
				Mel-spectrogram	89.72

Table 10 The effect learning rate and batch size in our CNN model

Learning rate	Batch size	Execution time (s)	Loss	Accuracy (%)
1	4	499.59	2.363	9.79
0.1	8	259.34	2.361	10.59
0.01	16	494.06	2.360	10.083
0.001	32	125.48	1.496	96.45
0.0001	**64**	**96.68**	**1471**	**99.083**
0.00001	128	103.5	1.472	98.83

Bolded value indicate the good experimental results

and North Shewa and tested with the Wollo dialect. The same procedure was applied by interchanging other dialects. From Table 9, we observed that the recognition performance of the model is greatly reduced through dialects by both MFCCs and Mel-spectrogram features. Therefore, we concluded that dialects have an impact on recognition accuracy.

We made an ablation study to choose an optimal batch size and learning rate as described in Table 10. Leaning rate and batch size are hyperparameters that are used to govern the pace at which an algorithm updates. Batch size is crucial since it influences both training time and model generalization. A lower batch size allows the model to learn from each individual example, but training takes longer.

A larger batch size trains a model faster, but the model may not capture the intricacies in the data. The learning rate controls the weight of the ANN concerning the loss gradient. The smaller the learning rate, it increases the training time. Therefore, we chose the optimal batch size and learning rate to get the appropriate performance of the model as shown in Table 10.

There are a few procedures involved in preprocessing speech data before it is fed into the neural network. To begin, we made an experiment as shown in Table 11 by downsampling all audio clips to a sampling rate of 8kHz to 24kHz. We observed that the higher sample rate increased the training time and did not have a significant effect on accuracy. Therefore, we selected the sample rate of 16kHz to prepare our dataset. Similarly, as shown in Table 12, the number of MFCCs also affects the training time of the model. As the number of MFCCs is increased, the training time of the models is also

Table 11 The effect of the sample rate in our CNN model

Sample rate	Trainable params	Execution time (s)	Loss	Accuracy (%)
8kHz	469,418	49.50	1.473	98.958
16 kHz	**993,706**	**96.68**	**1.471**	**99.083**
22.05 kHz	1,386,922	106.18	1.472	98.875
24 kHz	1,517,994	113.15	1.471	99.083

Bolded value indicate the good experimental results

Table 12 The effect of the number of MFCCs in our CNN model

No. of MFCCs	Trainable params	Execution time (s)	Loss	Accuracy (%)
13	**993,706**	**96.68**	**1.471**	**99.083**
15	993,706	89.99	1.472	98.95
20	1,452,458	116.71	1.473	98.75
25	2,369,962	162.57	1.473	98.83
30	2,828,714	187.64	1.473	98.85
35	3,287,466	229.63	1.472	98.92
40	3,746,218	251.54	1.473	98.75

Bolded value indicate the good experimental results

increased. Thus, for SDR, we used 13 MFCCs to speed up training time and to get better accuracy.

Conclusion

In this study, we have developed a new Amharic spoken digits dataset that contains 12,000 utterances. MFCCs and Mel-Spectrogram features were used to extract trainable features from wave signals. The performance of various classical supervised machine learning algorithms for Amharic spoken digits recognition was investigated. The recognition performance of these classical algorithms using the MFCCs feature is far better than the Mel-Spectrogram feature. Moreover, we have also proposed the Convolutional Neural Network (CNN) model to improve the recognition performance. The recognition performance of the proposed CNN using MFCCs and Mel-Spectrogram retains 99% and 98% accuracy, respectively. This result shows that the performance of the proposed CNN model is far superior to the baseline algorithms. Ethiopia has a lot of domestic languages that are widely spoken in different regions. Thus, the proposed deep learning model can also be applied to the development of spoken digits recognition, for other languages like Afaan Oromoo, Tigrigna, Somalia, etc. Moreover, the recognition performance of this system can be enhanced in the future by tuning model parameters and combining more than two feature extraction techniques instead of using a single feature extraction technique.

Abbreviations

SDR	Spoken digits recognition (SDR)
AmSDD	Amharic spoken digits dataset
AmSDR	Amharic spoken digits recognition
MFCCs	Mel frequency cepstral coefficients
LDA	Linear discriminant analysis
KNN	K-nearest neighbors

SVM	Support vector machine
RF	Random forest
CNN	Convolutional neural network

Acknowledgements
Authors would like to acknowledge and thanks to the participants in the collection of voice samples.

Author contributions
Conceptualization, T.A and G.M.B; methodology, T.A.A, and S.T.A; software, T.A.A and Z.C; validation, T.A.A and S.T.A; writing—original draft preparation, T.A.A., G.M.B. and H.L, and M.J; writing—review M.J and H.L.

Funding
This work was supported by the Postdoctoral Foundation of Zhejiang Normal University under Grant ZC304021941.

Availability of data and materials
We released the AmSDD on Kaggle for free download for research and academic purposes.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no conflict of interest.

Received: 24 May 2023 Accepted: 7 April 2024
Published online: 04 May 2024

References

1. Kaur AP, Singh A, Sachdeva R, Kukreja V. Automatic speech recognition systems: a survey of discriminative techniques. Multimed Tools Appl. 2022;82:1–33.
2. Aldarmaki H, Ullah A, Ram S, Zaki N. Unsupervised automatic speech recognition: a review. Speech Commun. 2022;139:76–91.
3. Deng L, Li X. Machine learning paradigms for speech recognition: an overview. IEEE Trans Audio Speech Lang Process. 2013;21(5):1060–89.
4. Kumar A, Verma S, Mangla H. A survey of deep learning techniques in speech recognition. In: 2018 international conference on advances in computing, communication control and networking (ICACCCN). IEEE; 2018. p. 179–85.
5. Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K. Speech recognition using deep neural networks: a systematic review. IEEE Access. 2019;7:19143–65.
6. Deng L, Hinton G, Kingsbury B. New types of deep neural network learning for speech recognition and related applications: an overview. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE; 2013. p. 8599–603.
7. Padmanabhan J, Johnson Premkumar MJ. Machine learning in automatic speech recognition: a survey. IETE Tech Rev. 2015;32(4):240–51.
8. Druzhkov P, Kustikova V. A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognit Image Anal. 2016;26:9–15.
9. Jiao L, Zhao J. A survey on the new generation of deep learning in image processing. IEEE Access. 2019;7:172231–63.
10. Rawat W, Wang Z. Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. 2017;29(9):2352–449.
11. Sharma G, Umapathy K, Krishnan S. Trends in audio signal feature extraction methods. Appl Acoust. 2020;158: 107020.
12. Ismail M, Memon S, Dhomeja LD, Shah SM, Hussain D, Rahim S, Ali I. Development of a regional voice dataset and speaker classification based on machine learning. J Big Data. 2021;8:1–18.
13. Korkmaz Y, Boyacı A. Hybrid voice activity detection system based on LSTM and auditory speech features. Biomed Signal Process Control. 2023;80: 104408.
14. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data. 2021;8:1–74.
15. LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackel L. Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems, vol. 2. 1989.
16. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, et al. Recent advances in convolutional neural networks. Pattern Recogn. 2018;77:354–77.
17. Sharmin R, Rahut SK, Huq MR. Bengali spoken digit classification: a deep learning approach using convolutional neural network. Procedia Comput Sci. 2020;171:1381–8.

18. Azim MA, Hussein W, Badr NL. Spoken arabic digits recognition system using convolutional neural network. In: Advanced machine learning technologies and applications: proceedings of AMLTA 2021. Springer; 2021. p. 164–72.
19. Chandio A, Shen Y, Bendechache M, Inayat I, Kumar T. AUDD: audio Urdu digits dataset for automatic audio Urdu digit recognition. Appl Sci. 2021;11(19):8842.
20. Tukeyev U, Karibayeva A, Zhumanov Z.h. Morphological segmentation method for Turkic language neural machine translation. Cogent Eng. 2020;7(1):1856500.
21. Abate ST, Menzel W, Tafila B, et al. An Amharic speech corpus for large vocabulary continuous speech recognition. INTERSPEECH. 2005;2005:1601–4.
22. Gereme F, Zhu W, Ayall T, Alemu D. Combating fake news in "low-resource" languages: Amharic fake news detection accompanied by resource crafting. Information. 2021;12(1):20.
23. Brhanemeskel GM, Abate ST, Ayall TA, Seid AM. Amharic speech search using text word query based on automatic sentence-like segmentation. Appl Sci. 2022;12(22):11727.
24. Leyew Z. The Amharic dialects revisited. From Beyond Mediterr Akten des. 2007;7:449–80.
25. Retta EA, Almekhlafi E, Sutcliffe R, Mhamed M, Ali H, Feng J. A new Amharic speech emotion dataset and classification benchmark. Trans Asian Low-Resour Lang Inf Process. 2022;22(1):1–22.
26. Korkmaz Y, Boyacı A. A comprehensive Turkish accent/dialect recognition system using acoustic perceptual formants. Appl Acoust. 2022;193: 108761.
27. Korkmaz Y, Boyacı A. Analysis of speaker's gender effects in voice onset time of turkish stop consonants. In: 2018 6th international symposium on digital forensic and security (ISDFS). IEEE; 2018. p. 1–5.
28. Oruh J, Viriri S, et al. Deep learning-based classification of spoken English digits. Comput Intell Neurosci. 2022. https://doi.org/10.1155/2022/3364141.
29. Mahalingam H, Rajakumar M. Speech recognition using multiscale scattering of audio signals and long short-term memory of neural networks. Int J Adv Comput Sci Cloud Comput. 2019;7(2):12–6.
30. Nasr S, Quwaider M, Qureshi R. Text-independent speaker recognition using deep neural networks. In: 2021 international conference on information technology (ICIT). IEEE; 2021. p. 517–21.
31. pannous: Pannous/TensorFlow-speech-recognition. 2014. http://github.com/pannous/tensorflow-speech- recognition.
32. A free audio dataset of spoken digits. Think MNIST for audio. 2014. https://github.com/Jakobovski/free-spoken-digit- dataset.
33. Sarma P, Sarmah S, Bhuyan M, Hore K, Das P. Automatic spoken digit recognition using artificial neural network. Int J Sci Technol Res. 2019;8(12):1400–4.
34. Taufik D, Hanafiah N. Autovat: an automated visual acuity test using spoken digit recognition with MEL frequency cepstral coefficients and convolutional neural network. Procedia Comput Sci. 2021;179:458–67.
35. Wazir ASMB, Chuah JH. Spoken Arabic digits recognition using deep learning. In: 2019 IEEE international conference on automatic control and intelligent systems (I2CACIS). IEEE; 2019. p. 339–44.
36. Zerari N, Abdelhamid S, Bouzgou H, Raymond C. Bidirectional deep architecture for Arabic speech recognition. Open Comput Sci. 2019;9(1):92–102.
37. Hasnain S, Awan MS. Recognizing spoken Urdu numbers using Fourier descriptor and neural networks with matlab. In: 2008 second international conference on electrical engineering. 2008; IEEE. p. 1–6.
38. Ali H, Jianwei A, Iqbal K. Automatic speech recognition of Urdu digits with optimal classification approach. Int J Comput Appl. 2015;118(9):1–5.
39. Gupta A, Sarkar K. Recognition of spoken Bengali numerals using MLP, SVM, RF based models with PCA based feature summarization. Int Arab J Inf Technol. 2018;15(2):263–9.
40. Paul B, Bera S, Paul R, Phadikar S. Bengali spoken numerals recognition by MFCC and GMM technique. In: Advances in electronics, communication and computing: select proceedings of ETAEERE 2020. Springer; 2021. p. 85–96.
41. Das S, Yasmin MR, Arefin M, Taher KA, Uddin MN, Rahman MA. Mixed Bangla–English spoken digit classification using convolutional neural network. In: Applied intelligence and informatics: first international conference, AII 2021, Nottingham, UK, July 30–31, 2021, proceedings 1. Springer; 2021. p. 371–83.
42. Dhandhania V, Hansen JK, Kandi SJ, Ramesh A. A robust speaker independent speech recognizer for isolated Hindi digits. Int J Comput Commun Eng. 2012;1(4):483.
43. Zada B, Ullah R. Pashto isolated digits recognition using deep convolutional neural network. Heliyon. 2020;6(2):03372.
44. Musaev M, Khujayorov I, Ochilov M. Image approach to speech recognition on CNN. In: Proceedings of the 2019 3rd international symposium on computer science and intelligent control. 2019. p. 1–6.
45. Renjith S, Joseph A, KK AB. Isolated digit recognition for Malayalam—an application perspective. In: 2013 international conference on control communication and computing (ICCC). IEEE; 2013. p. 190–3.
46. Dalsaniya N, Mankad SH, Garg S, Shrivastava D. Development of a novel database in Gujarati language for spoken digits classification. In: International symposium on signal processing and intelligent recognition systems. Springer; 2020. p. 208–19.
47. Robert J. pydub. 2011. https://github.com/jiaaro/pydub.
48. Shrawankar U, Thakare VM. Techniques for feature extraction in speech recognition system: a comparative study. arXiv preprint. 2013. arXiv:1305.1145.
49. Dave N. Feature extraction methods LPC, PLP and MFCC in speech recognition. Int J Adv Res Eng Technol. 2013;1(6):1–4.
50. Alías F, Socoró JC, Sevillano X. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. Appl Sci. 2016;6(5):143.
51. Gupta S, Jaafar J, Ahmad WW, Bansal A. Feature extraction using MFCC. Signal Image Process Int J. 2013;4(4):101–8.
52. Al Bashit A, Valles D. A mel-filterbank and MFCC-based neural network approach to train the Houston toad call detection system design. In: 2018 IEEE 9th annual information technology, electronics and mobile communication conference (IEMCON). IEEE; 2018. p. 438–43.

53. McFee B, Raffel C, Liang D, Ellis DP, McVicar M, Battenberg E, Nieto O. librosa: Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference, vol. 8. 2015. p. 18–25.
54. Tharwat A, Gaber T, Ibrahim A, Hassanien AE. Linear discriminant analysis: a detailed tutorial. AI Commun. 2017;30(2):169–90.
55. Cutler A, Cutler DR, Stevens JR. Random forests. In: Ensemble machine learning: methods and applications. Birmingham: Packt Publishing; 2012. p. 157–75.
56. Apicella A, Donnarumma F, Isgrò F, Prevete R. A survey on modern trainable activation functions. Neural Netw. 2021;138:14–32.
57. Moolchandani D, Kumar A, Sarangi SR. Accelerating CNN inference on ASICs: a survey. J Syst Architect. 2021;113: 101887.
58. Audacity: open source, cross-platform audio software. https://www.audacityteam.org/. Accessed 1 Nov 2022.
59. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. Pytorch: an imperative style, high-performance deep learning library. In: Advances in neural information processing systems, vol. 32. 2019.
60. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
61. Erickson BJ, Kitamura F. Magician's corner: 9. Performance metrics for machine learning models. Radiol Artif Intell. 2021;3(3): 200126.
62. Aggarwal A, Sahay T, Chandra M. Performance evaluation of artificial neural networks for isolated Hindi digit recognition with LPC and MFCC. In: 2015 international conference on advanced computing and communication systems. IEEE; 2015. p. 1–6.
63. Tailor JH, Rakholia R, Saini JR, Kotecha K. Deep learning approach for spoken digit recognition in Gujarati language. Int J Adv Comput Sci Appl. 2022;13(4)424–429.
64. Silva DF, de Souza VM, Batista GE, Giusti R. Spoken digit recognition in Portuguese using line spectral frequencies. In: Advances in artificial intelligence–IBERAMIA 2012: 13th Ibero-American conference on AI, Cartagena de Indias, Colombia, November 13–16, 2012. Proceedings 13. Springer; 2012. p. 241–50.
65. Nisar S, Shahzad I, Khan MA, Tariq M. Pashto spoken digits recognition using spectral and prosodic based feature extraction. In: 2017 ninth international conference on advanced computational intelligence (ICACI). IEEE; 2017. p. 74–8.

Publisher's Note