School of Computer Science and Engineering

Natural Language Processing Project Report

On

Voice Controlled Tic-Tac-Toe Game

Submitted by Team 5,

Akshay Joshi (01fe20bcs102)

Kartik Kalal (01fe20bcs116)

Dhiraj Bhandare (01fe20bcs117)

Vaishnavi Patil (01fe20bcs196)

Under the guidance of,

Ms. Priyadarshini Patil

Contents

1. Introduction

Natural Language Processing (NLP) helps computers learn, understand, and produce content in human or natural language. Speech recognition can process human speech into a written format. Speech recognition in Python works with algorithms that perform linguistic and acoustic modeling. Acoustic models represent the relationship between linguistic units of speech and audio signals. In language models, sounds are matched with word sequences to distinguish between words that sound similar. Speech recognition starts by taking the sound energy produced by the person speaking and converting it into electrical energy with the help of a microphone. It then converts this electrical energy from analog to digital, and finally to text.

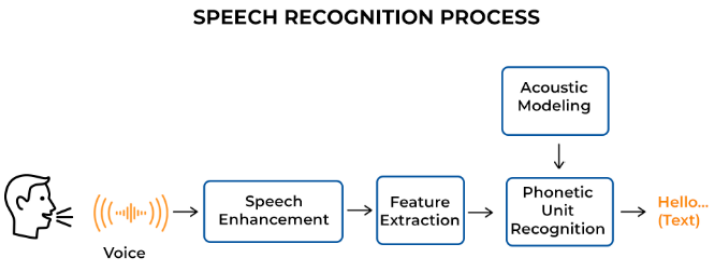


Fig 1. Speech recognition pipeline.

In this proposed work, we have tried to implement a system that makes use of human speech to play a game of Tic-Tac-Toe. Tic-tac-toe is played on a three-by-three grid by two players, who alternately place the marks X and O in one of the nine spaces in the grid. There is no universally-agreed rule as to who plays first, but in this proposed work the convention that X plays first is used. This game is designed by applying speech recognition features with deep convolutional neuro-learning.

Fig 2. Representation of Tic-Tac-Toe.

2. Literature Survey

2.1. Spoken Digit Recognition (Speech Recognition)

In this paper, the authors have tried to achieve perfect accuracy for spoken digit recognition. The dataset used contains 30,000 labelled audio clips of digits from 0 to 9. The authors have tested the effectiveness of linear neural network and convolutional neural network on the audio clips. Because the linear neural networks were underperforming, a deep convolutional neural network was implemented that created a spectrogram of the audio clips and inputs them as 2-dimensional matrices into the network. The last layer of the network is pushed through a SoftMax activation function to output the 10 probabilities that the audio is each of the digit possibilities.

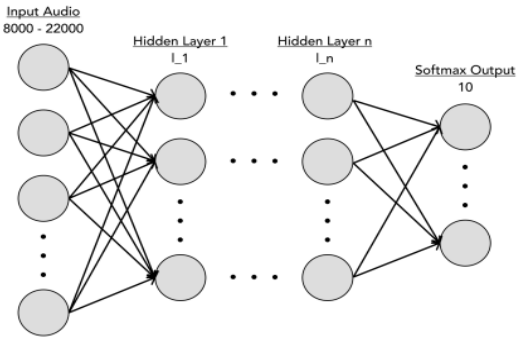


Fig 3. Neural network model based on survey paper 2.1.

This work resulted with 99.6% training accuracy and 99.3% test accuracy.

Channels	Kernels	FC Layers	Epochs	Train Accuracy	Dev Accuracy
1 - 5 - 10	5 - 5	Flat - 1000 - 10	50	99.6%	99.3%
1 - 10 - 20	5 - 5	Flat - 1000 - 10	50	99.6%	99.3%
1 - 5 - 10	9 - 5	Flat - 1000 - 10	50	98.6%	98.3%
1 - 5 - 10	5 - 5	Flat - 5000 - 1000 - 10	50	99.7%	98.6%
1 - 10 - 20	5 - 5	Flat - 5000 - 1000 - 10	30	99.1%	99.0%

Fig 4. Results obtained from survey paper 2.1.

2.2. Automatic spoken digit recognition using artificial neural network

In this paper, the authors have built a system that will recognize digits spoken in English. The system was trained on voice samples of the people of the Northeastern region of India. The system uses Linear Prediction Coefficient (LPC), which helps to recover the corrupted segments of the input speech at the receiving end making it a good feature extraction method and Principal Component Analysis (PCA) for signal analysis.

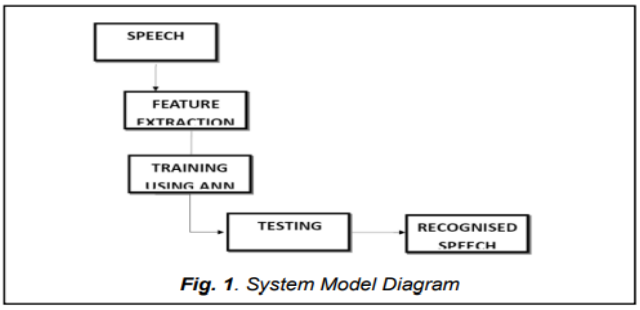


Fig 5. Flowchart of the model used in survey paper 2.2.

The system is tested against the voice signal of around 50 persons and the system gives approximately 82% accuracy. It was found that with the increase of number of samples the accuracy of the recognition rate increases.

VOICE SAMPLES OF SIZE 50

Voice samples (10)	Recognition Rate (%)	Un-Recognized Rate (%)
Zero	82	18
One	88	12
Two	80	20
Three	88	12
Four	82	18
Five	80.8	19.2
Six	82.6	17.4
Seven	83	17
Eight	86	14
Nine	86	14

Fig 6. Results based on survey paper 2.2.

2.3. Speech Based Voice Recognition System for Natural Language Processing

This paper explains about the speech-based voice authentication system for Tamil language that has two major phases, feature excerption phase and feature matching phase. In the feature excerption phase, the system extracts the MFCC features from the voice input sample. In the feature matching module, it identifies the Tamil words and the user using the Dynamic Time Warping algorithm that computes the warping distance between two-time sequences. The two-time series is similar when the warping distance between them is very small. Thus, the system creates the voice password using Tamil words.

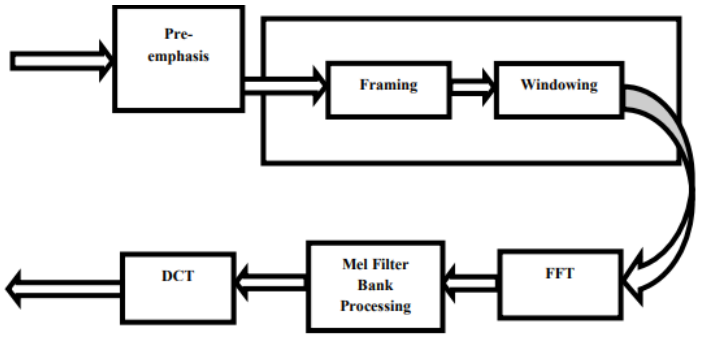


Fig 7. Pipeline followed by survey paper 2.3.

2.4. Speech Recognition Systems - A Comprehensive Study of Concepts and Mechanism

This paper reviews various aspects related to Speech Recognition Technology and the system implementing this technology to engineer Speech Recognition Systems. There have been many approaches adapted to implement speech recognition technology in machines and systems, out of which Dynamic Type Warping (DTW) and Hidden Markov Model (HMM) have been the center of attraction. DTW is an algorithm used for pattern matching between the two signals that may vary in speed and time, while HMM is a model which is used to study hidden or unobserved states. DTW is widely used in areas of speaker recognition and online signature recognition and partially used in shape matching

application. HMM is widely used in areas of speech conversion, gestures and tagging of POS (Part of Speech). However, there are many challenges in Speech Recognition such as Noisy Environment, Intensive use of computer power, Accent, Speed of speech, Recognition of punctuation marks, Homophones etc. To overcome these challenges there are various tools such as Voice activity detector and AURORA experimental framework. Technological advances in computation has led the technology of Speech Recognition reach the state where situations are far better as they were used to be years back and definitely in the coming few years the world is about to experience much better language understanding by the machines.

2.5. Speech Recognition using Machine Learning

In this paper, the authors have come up with a system that is able to recognize speech and convert input audio into text. It also enables a user to perform file operations like Save, Open or Exit from voice-only input. The system is also able to recognize the human voice as well as audio clips, and translate between English and Hindi.

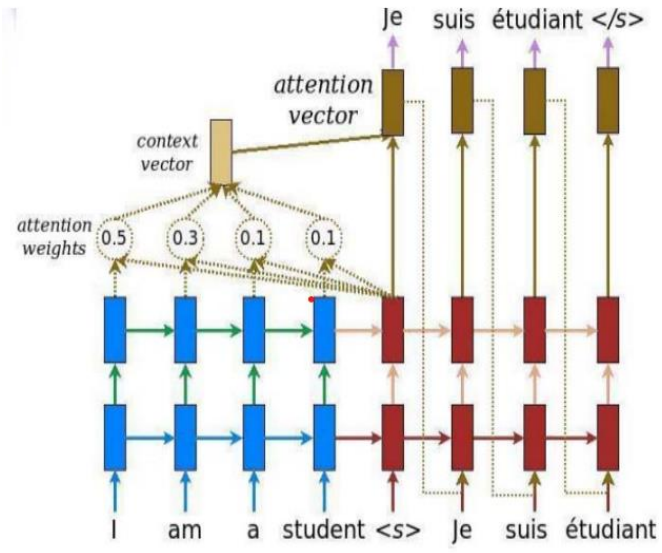


Fig 8. Architecture followed by survey paper 2.5.

The architecture used is a conventional encoder-decoder units made up of RNNs with attention-based mechanism. The constraint of the encoder–decoder model, attention, is proposed to encrypt the input sequence to one fixed length vector from which each output time stage is decoded. With long sequences, focus is proposed as a strategy to both align and interpret, and this problem is believed to be more of a concern. Instead of encoding the input sequence into a single fixed-context vector, the attention model produces a context vector that is filtered independently for each output time step. The approach is extended, as with the encoder–decoder text, to a machine translation problem, and uses GRU units rather than LSTM memory cells [9]. In this case, bidirectional input is used where both forward and backward input sequences are given, which are then concatenated before being passed to the decoder. The input is positioned into an encoder model that gives us the output of the form encoder and the hidden shape state encoder.

3. Dataset Description

The dataset used in our work is Audio MNIST dataset. This dataset consists of 30,000 1-second recordings from 60 different speakers saying a digit from 0 to 9. The data comes along with the gender, age, and nationality of each individual speaker. There is one directory per speaker holding the audio recordings. The recordings were captured at a sample rate of 48kHz and trimmed to minimize silence. Each of these audio clips is associated with the correct value that was spoken in the clip. All audio clips are down-sampled to a sampling rate of between 8kHz and 22kHz. Since speech is low bandwidth (between 100Hz - 8kHz), 8kHz would probably be sufficient. All audio clips are zero padded to make them exactly 1 second so that they are equal inputs into the neural network.

4. Proposed Methodology

4.1. Feature Extraction

We have experimented with two different ways of inputting the audio file into the convolutional neural network. The first method is to extract the amplitudes of the audio clip into a large array. Below is the visual waveform of someone saying "five" after being pre-processed:

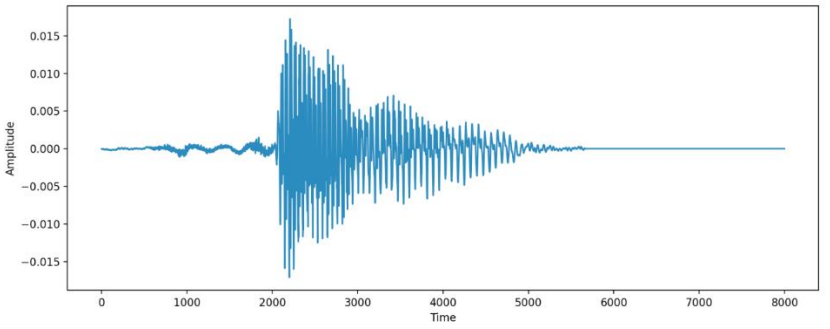


Fig 9. Visual waveform of an audio clip.

The second method is to convert the audio file into a spectrogram of frequencies. Below is a spectrogram of someone saying "five" after being pre-processed:

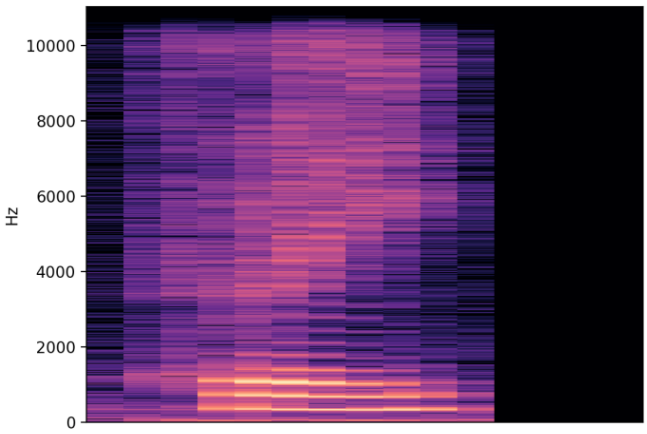


Fig 10. Spectrogram representation of an audio clip.

With 30,000 audio clips in the dataset, we went with a 90-10 random split for training and validation sets respectively.

4.2. Convolutional Neural Network

Convolutional Neural Network (CNN) is applied as advanced deep neural networks to classify each spoken digit from our pooled data set as a multi-class classification task. Below is the proposed deep neural network:

```
SpokenDigitModel(
  (network): Sequential(
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU()
    (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (9): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (10): ReLU()
    (11): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (12): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU()
    (14): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (15): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (16): ReLU()
    (17): AdaptiveAvgPool2d(output_size=1)
    (18): Flatten(start_dim=1, end_dim=-1)
    (19): Linear(in_features=256, out_features=128, bias=True)
    (20): ReLU()
    (21): Linear(in_features=128, out_features=64, bias=True)
    (22): ReLU()
    (23): Linear(in_features=64, out_features=10, bias=True)
    (24): Softmax(dim=1)
  )
)
```

Fig 11. Summary of the proposed neural network.

CNN is used to train and test our data. This neural network is made up of 17 layers out of which 6 are weight layers. Below is an explanation of each of the layers used in the neural network.

Convolution layer:

- Filters are included to find features of the audio.
- The filter consists of small kernels (number of kernels), with one bias per filter.
- For every value of feature map, ReLU activation is applied to introduce non-linearity in the model.

Pooling layer:

- Pooling layer is used to extract maximum or average area.
- It follows sliding window concept.
- Used to reduce the dimensionality.

4.3. Keyword Spotting Service

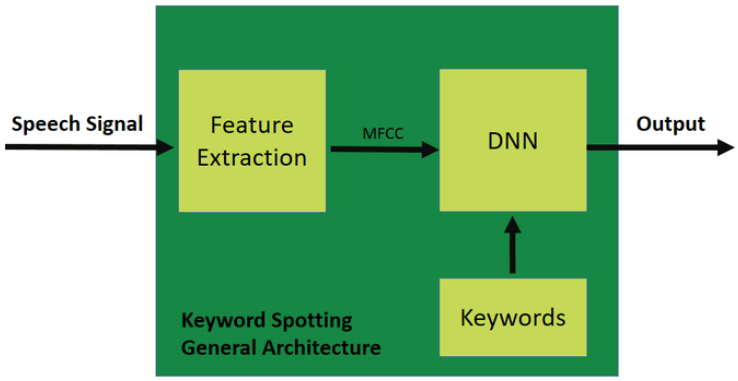


Fig 12. KSS Architecture

Keyword Spotting Services (KSS) is the process of recognizing predefined words from a speech signal. The process involves converting speech signals into a feature vector that can be used as an input to a model. The model then compares the input to the predefined words and detects their presence.

4.4. Training

The model was trained for 16 epochs with learning rate of 0.001 on Audio MNIST dataset. Given training input x, and SoftMax output y, the network optimizes the Multi-Label Soft Margin Loss function, defined as:

$$\text{loss}(x, y) = -\frac{1}{C} * \sum_i y[i] * \log((1 + \exp(-x[i]))^{-1}) + (1 - y[i]) * \log\left(\frac{\exp(-x[i])}{(1 + \exp(-x[i]))}\right)$$

After training the model, the calculated training accuracy was found out to be 92.85% with a loss value of 1.5318.

4.5. Tic-Tac-Toe

The user can select any of the 3 modes provided:

- Human vs Computer
- Human vs Human
- Human vs Smart-Computer

Once the mode has been selected, the game begins. Since tic-tac-toe is a 3 x 3 grid, the user must give the grid number as input through voice where he wants “X” or “O” to be placed. This voice is sent as an audio file to the model predictor. The predictor will predict the value of the grid number and places “X” or “O” accordingly. The turn is passed to the other player and the process is repeated until the game ends. A well-known game playing algorithm called MiniMax algorithm has been used for decision-making and game theory. It delivers an optimal move for the player, considering that the competitor is also playing optimally.

In this algorithm, two players play the game; one is called ‘MAX’ and the other is ‘MIN.’ The goal of players is to minimize the opponent’s benefit and maximize self-benefit. The MiniMax algorithm conducts a depth-first search to explore the complete game tree and then proceeds down to the leaf node of the tree, then backtracks the tree using recursive calls. This allows the player to find the best optimal move in the game.

5. Results

The training accuracy was found to be 92.85% with a loss value of 1.5318 whereas the testing accuracy was found to be 98.32% with a loss value of 0.1889.

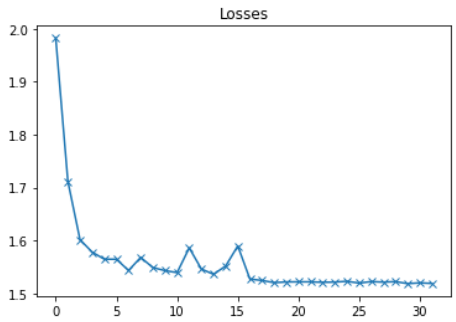


Fig 13. Epoch vs Loss Function

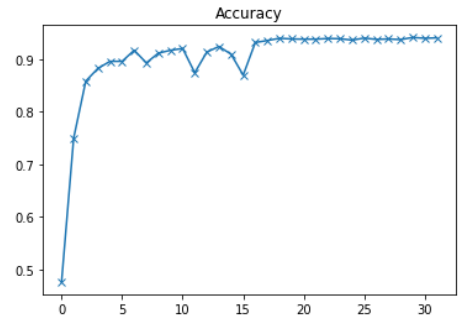
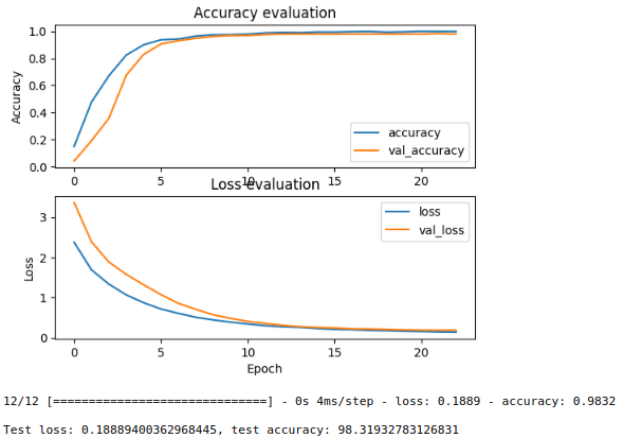


Fig 14. Epoch vs Training Accuracy




```
12/12 [==============================] - 0s 4ms/step - loss: 0.1889 - accuracy: 0.9832
Test loss: 0.18889400362968445, test accuracy: 98.31932783126831
```

Fig 15. Test dataset evaluation

6. Conclusion

Our data consists of ten spoken words from 60 people, each word represents one label from our ten labels that are applied for supervised learning of our suggested deep neural model. The suggested CNN structure has 17 layers out of which 6 are weight layers. In our model, audio files have been sent as input directly to the designed network structure that contained multilevel learning procedure to achieve the model multi-classification task. The experiment results show that the deep neural networks have ability to solve speech recognition challenges.

CNN returned an acceptable performance for spoken digit recognition task. The model returned the training accuracy as 92.85% with a loss value of 1.5318. A widely used game-playing algorithm known as MiniMax algorithm has been used for optimal decision-making in Tic-Tac-Toe. The model performance could be increased and achieve better classification accuracy when the model is trained for larger epochs and appropriate regularization techniques.

References

- Spoken Digit Recognition (Speech Recognition): http://cs230.stanford.edu/projects_fall_2020/reports/55617928.pdf
- Automatic spoken digit recognition using artificial neural network: http://www.ijstr.org/final-print/dec2019/Automatic-Spoken-Digit-Recognition-Using-Artificial-Neural-Network.pdf
- Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals: https://arxiv.org/abs/1807.03418
- Speech Based Voice Recognition System for Natural Language Processing: https://www.researchgate.net/publication/284107344_Speech_Based_Voice_Recognition_System_for_Natural_Language_Processing
- Audio MNIST dataset: https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist
- Spoken digit dataset: https://www.kaggle.com/datasets/divyanshu99/spoken-digit-dataset
- MiniMax algorithm: https://levelup.gitconnected.com/minimax-algorithm-explanation-using-tic-tac-toe-game-22668694aa13
- Keyword Spotting: https://arxiv.org/abs/2111.10592