ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 10| Issue: 4| April 2024|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2024: 8.402 || ISI Value: 1.188

PRONUNCIATION LEARNING USING AUTOMATIC SPEECH RECOGNITION

Rohit Sharma[1], Amey Menon[2], Sahil Khairnar[3] Sanket Bhagat[4], Shubham Rawat[5]

[1]*Professor, Information Technology Engineering, PCEIT , New Panvel Navi Mumbai*
[2]*B.E in Information Technology Engineering, PCEIT, New Panvel Navi Mumbai*
[3]*B.E in Information Technology Engineering,PCEIT, New Panvel Navi Mumbai*
[4]*B.E in Information Technology Engineering,PCEIT, New Panvel Navi Mumbai*
[5]*B.E in Information Technology Engineering,PCEIT, New Panvel Navi Mumbai*

Article DOI: https://doi.org/10.36713/epra16634
DOI No: 10.36713/epra16634

ABSTRACT

Due to limited interaction between the teacher and the student, learning a foreign language can be difficult. Language learning, in contrast to most other courses, necessitates oral practice and interactive corrective feedback, which may be unavailable with little study material and time for interactions. It might not be possible for the teacher to give each student their whole attention in a classroom setting. By enabling better and more flexible work and digitizing study materials utilizing cutting- edge signal processing techniques, modern computer technology can enhance language acquisition. The project offers an online pronunciation learning tool that follows the listen and repeat method.In order to enable foreign language learners to practice their abilities remotely and even without the teacher's presence, it offers an interactive interface. The underlined text will be phonetically transcriptioned by the use of automatic speech recognition software to record and process human speech. The teacher will compare it to the prompt text. Students can hear teacher-generated speech as feedback and adjust their pronunciation until it is recognized correctly.

KEYWORDS: E-*Learning, Pronunciation learning, Automatic Speech Recognition*

1.INTRODUCTION

In the modern world, learning a new language and giving it the proper pronunciation is crucial. It is essential to switch to computer- based learning in light of the benefits of digitization. studying communication skills on a computer can sometimes be more successful than studying from an instructor. The application of pronunciation-based learning is highlighted in this project. There are instances when students are unable to ask questions or when the teacher is unable to address every student's concern. Students can watch or record their pronunciation and compare it to a teacher's pronunciation with the aid of e-learning platforms. When compared to other people, those who are afraid of learning a new language frequently have greater difficulties. Since learning a new language requires stepping outside of their comfort zone, they are afraid they won't succeed.

They are also unsure about the variations in writing, sounds, etc. As a result, one should refrain from delving too far before beginning. Acquiring focus is necessary when learning a language in order to recognize and comprehend the subtle distinctions that distinguish each language. In recent years, a significant amount of research has been conducted in the field of speech signal processing. Particularly, the subject of automated speech recognition (ASR) technologies has seen a rise in interest. ASR started off as basic systems that could only recognize a few number of sounds and has now developed into complex systems that can understand and speak genuine language.

2. PROBLEM STATEMENT

Precise pronunciation is essential for effective communication throughout language learning. But whether learning a foreign language or a native tongue, learners usually find it difficult to understand correct pronunciation.The following are some of the challenges associated with learning any new language (pronunciation):

- Study material
- the need for a teacher
- time constraints for instruction
- variations in instructor pronunciation
- and the expense of hiring a teacher
- monitoring and displaying the student's development over time

3. STEPS INVOLVED

The following are the steps that our project entails:
Step 1: Choose a random audio file from the dataset in step one. Step 2: View the audio spectrogram produced with Python libraries. Step 3: Use the WebRTC module in JavaScript to record user audio. Step 4: To create an audio spectrum, recorded audio will be sampled and plotted.
Step 5: A CNN model is used to match the recorded audio with theaudio in the dataset.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 10| Issue: 4| April 2024|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2024: 8.402 || ISI Value: 1.188

Step 6: Predicting the text for the captured audio.
Step 7: A score representing the maximum matching probability is shown.

4. BLOCK DIAGRAM

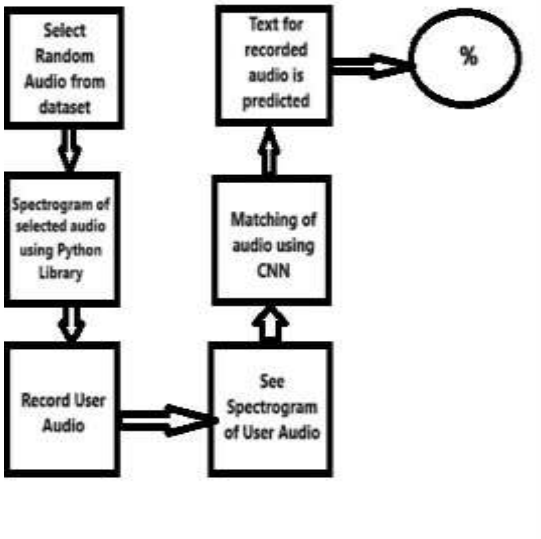

Figure 1: Block Diagram

5.COMPONENTS AND SOFTWARES

The components and softwares we have used in this project are as follows:

- HTML
- CSS
- Bootstrap
- JavaScript
- WebRTC
- Flask
- Librosa
- Plotly
- Python Script
- Tensorflow
- Keras
- Numpy
- Dataset

5.1. Dataset

The Speech Command dataset from Google is the one being used. This publicly available, free dataset is suitable for novice users. We used the audio files for the digits 0 to 9 out of 65,000 one-second long utterances of 30 short syllables. The dataset was donated by the public.With the use of this dataset, developers may create straightforward voice interfaces for applications that use simple phrases like "no," "yes," and numbers like "d." The data production infrastructure is publicly available and utilized by the broader community to generate new iterations, primarily encompassing languages and their applications.If your speech patterns are included in the dataset, the outcome will be determined by those patterns. Since speech recognition in commercial systems is more complicated than in this instructional example, speech patterns might not be flawless.

We will undoubtedly continue to see expansions and enhancements as additional accents and variations are added to the dataset and as users contribute better models to TensorFlow. Furthermore, any type of speech dataset or training dataset can be used in our research.

6.WORKING

- The user must utilize Flask and Data to listen to the audio file that is already saved in the web application's backend Data in JSON format.
- The user is able to view the sound's spectrogram that he must pronounce.
- The user gets the option to record his own voice pronouncing words that he has previously heard.
- The user must then download and upload this recorded audio file.
- Python libraries such as Librosa, Numpy, and Matplotlib will then be used to process the user's audio, and the resulting Spectrogram will be delivered to the user as an output.
- We have created a CNN model for a probabilistic output that receives user audio as input, transforms it into a spectrogram, and outputs the word that has the highest chance of matching the audio in the dataset.
- At last, the percentage of matching between the audio files is obtained.

7. IMPLEMENTATION

In order to carry out our project, we went through the following procedure:

7.1 Software

The first step in audio classification tasks is to identify the class to which a sound sample belongs from a list of potential classes. The following is the training data for Speech-to-Text problems: Spoken word audio clips (X) are input features. Labels: a written transcript of the spoken word for the target (y).

7.2 Data Pre-Processing

Conventional audio processing techniques are no longer necessary, and standard data preparation may be relied upon without the need for laborious human feature development. Any automatic speech recognition system starts with the process of extracting features, or figuring out which parts of the audio signal are relevant for linguistic content detection and which ones should be ignored, like emotion, background noise, and other distractions. Unprocessed audio data is not something we work with. Rather, it is common practice to convert auditory input into images, which are subsequently handled by a standard CNN architecture. This is achieved by generating spectrograms from the audio..One common method is to convert auditory inputs into visual representations, usually pictures, and then process those images using a traditional Convolutional Neural Network (CNN)

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 10| Issue: 4| April 2024|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2024: 8.402 || ISI Value: 1.188

architecture. The audio signals are transformed into spectrograms, a visual depiction that records the frequency content of the audio across time, in order to accomplish this translation.

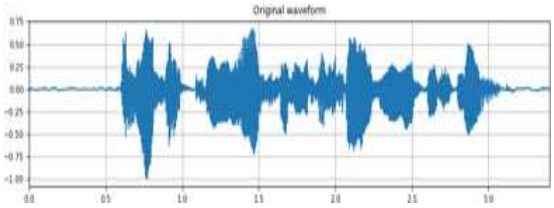

Figure 2: Audio Waveform

7.3. Spectrogram

Any signal can be broken down into its component frequencies using Fourier transforms, such as the Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), and Short Time Fourier Transform (STFT), which produce spectrograms. The most popular library for creating spectrograms is Librosa. Furthermore, a CNN- based model designed to handle images would benefit greatly from the input of a spectrogram, as it is an image.

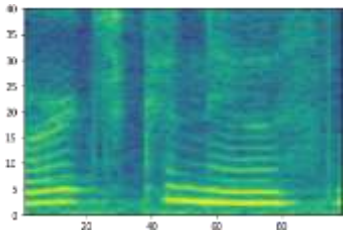

Figure 3: Spectrogram

7.4 MFCC

A variety of audio deep learning applications can benefit from the use of Mel Spectrograms. However, MFCC (Mel Frequency Cepstral Coefficients) might be better in scenarios involving human voice, like Automatic Speech Recognition. These take Mel Spectrograms and process them through a few further processes. The most common human speech frequencies are represented in the frequency bands that are selected from a compressed version of the Mel Spectrogram. The most crucial audio characteristics for capturing the sound core quality are those that the MFCC recovers from a far smaller collection.

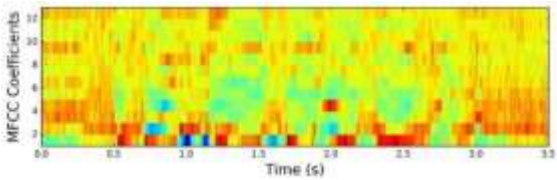

Figure 4: MFCC

7.5 Loading Dataset

As indicated in Table, we randomly split the dataset into training and test sets in an 80:20 ratio. Eighty percent of the dataset is used to train the model, while twenty percent is utilized for testing. Cross- validation uses 20% of the training set as well.

7.6 Data Augmentation

Increasing the diversity of your dataset by artificial means is a common tactic, particularly in cases where you don't have enough data. We accomplish this by slightly altering the present data samples. Both the final spectrogram and the raw audio used to create it can be enhanced with this technique. Generally, greater results are obtained by increasing the spectrogram's resolution.
Spectrograms react differently to picture modifications than do photos. For instance, a horizontal flip or rotation would have a big impact on the spectrogram and the music it represents. Rather, we use the Spec Augment method, which involves obstructing some regions of the spectrogram. The two masks that are most frequently employed are the time mask and the frequency mask. A variety of techniques, such as time stretching, pitch shifting, time shifting, adding noise, and more, can be used to enhance raw audio.

7.7 CNN Model Architecture

Three convoluted layers make up the CNN model utilized in this project. Each convolutional layer is succeeded by a max-pooling and batch-normalization layer.
The model ultimately consists of three fully connected layers. Ten pieces will make up the final layer, or the Softmax layer, since there are ten classes available for recognition.
There are several learnable parameters in each layer. The convolutional layer's parameter count is equal to ((filter width x filter height x number of filters in the preceding layer) + bias term) x number of filters.
Four times the number of filters in the preceding layer is the number of parameters in the batch nor
malization layer
. Since there is no learning taking place at the pooling layer, it has no parameters. All it does is dimension reduction.
(Previous layer neurons * current layer neurons) + (bias term x current layer neurons) equals the number of
parameters for a fully connected layer.) The bias term that we have adopted is 1.
Consequently, the quantity of parameters in every layer can be computed as follows:

- First Convolutional Layer = ((3 x 3) + 1) x 64 = 640
- Batch Normalization Layer 1 = 4 x 64 = 256
- Max Pooling Layer 1 = 0
- Second Convolutional Layer = ((3 x 3 x 64) + 1) x 32 =18464
- Batch Normalization Layer 1 = 4 x 32 = 128
- Max Pooling Layer 2 = 0
- Third Convolutional Layer = ((2 x 2 x 32) + 1) x 32 = 4128
- Batch Normalization Layer 1 = 4 x 32 = 128
- Max Pooling Layer 3 = 0
- First Fully connected Layer = 0
- Second Fully connected Layer = ((64 x 160) + (1 x 64) = 10304
- Third Fully connected Layer = ((10 x 64) + (1 x 10) = 650

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 10| Issue: 4| April 2024|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2024: 8.402 || ISI Value: 1.188

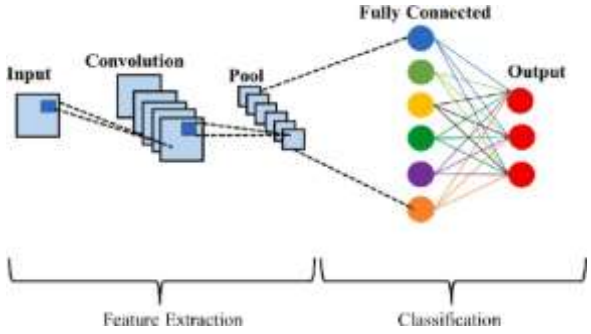

Figure 6: CNN

7.8. Network Hyper-Parameters

- Epochs = 40
- Batch Size = 32
- Learning rate = 0.0001
- Patience = 5
- Kernel width = 2 to 3
- Number of filters per kernel = 32 to 64
- Number of nodes in hidden layers = 10 to 160

7.9 Training

Forty epochs were used to train the model. A batch size of 32 is employed during the five epochs of training. To prevent the model from being overfit, an early training stop was used. The model's execution is halted if it attempts to exceed t. During the learning process, the Adam optimization for stochastic gradient descent is applied at a learning rate of 0.0001

8. RESULTS AND CONCLUSION

We discovered the following project outputs and findings while implementing our project:

8.1. Result Analysis

A validation accuracy of 95.30% and a train accuracy of 97.34% were attained by the CNN model. Plots of accuracy versus epoch and loss versus epoch are displayed in the figure

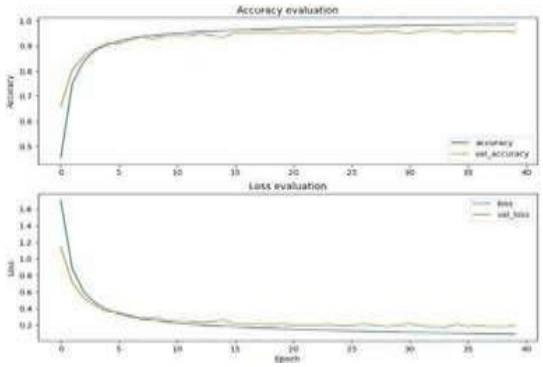

Figure 7: Accuracy

8.2 Project Outcomes

Our project's two primary results are:

- provides a spectrogram as the user's audio input's output. Gives
- assigns a score depending on the model's prediction for the audio input.

Figure 8: User Interface

9. REFERNCES

1. *Yi-Chin Kao; Chung-Ting Li; Tzu-Chiang Tai; Jia-Ching Wang, "Emotional Speech Analysis Based on Convolutional Neural Networks", IEEE,2021 https://ieeexplore.org/document/9411528/keywords#keyword*
2. *D.G. Kodagoda; K.G.R.U Ishara; R.M.R.P Kumara;W.A.D.T Dilshan; Lakesha Weerasinghe; Nadeesa Premadasa "An Interactive E-Learning Tool",IEEE,2022. https://ieeexplore.ieee.org/document/4283789*
3. *Shehu Mohammed Yusuf; E. A. Adedokun; M. B. Muazu;J. Umoh; Ahmed Abdul Ibrahim "RMWSaug: Robust Multi-window Spectrogram Augmentation Approach for Deep Learning based Speech Emotion Recognition", IEEE,2021 https://ieeexplore.ieee.org/document/9598956*
4. *Wei-Cheng Lin; Dimitra Emmanouilidou " Toxic Speech and Speech Emotion Investigations of Audio based Modelling and Intercorrelations" 2022 https://ieeexplore.ieee.org/document/9909856*
5. *Oussama Mounnan; Otman Manad; Larbi Boubchir; Abdelkrim El Mouatasim; Boubaker Daachi "Deep Learning based Speech Recognition System using Blockchain for Biometric Access Control" 2022. https://ieeexplore.ieee.org/document/10062921*
6. *Ashwini Jadhav, Ritesh Ajoodha, " The use of Automatic Speech Recognition in Education for Identifying Attitudes of the Speakers",IEEE, 2020https://ieeexplore.ieee.org/document/9411528*