DSE 3159 Deep Learning

Gender recognition using speech signals

Sahil Patil

Data Science and Computer applications

200968154

Manipal Institute of Technology

Abstract

The objective is to classify genders (Male/Female) using speech signals from audio files. Gender identification based on the voice of a speaker consists of detecting if a speech signal is uttered by a male or a female. Automatically detecting the gender of a speaker has several potential applications. In the context of Automatic Speech Recognition, gender dependent models are more accurate than gender independent ones.

Metadata

The dataset for this project is taken from VoxCeleb. It consists of two folders males and females, which have over 3000 audio files containing compressed WAV (waveform audio) files to M4A (MPEG-4 audio) format having lower audio quality. M4A file extension usually contain digital audio stream encoded with AAC.

Link: https://www.robots.ox.ac.uk/~vgg/data/voxceleb

Techniques

Two main approaches can be used for gender identification. One approach is to use gender dependent features, such as the pitch. The other approach is to use a general pattern recognition approach based on general speech features such as the Mel Frequency Cepstral Coefficients (MFCC).

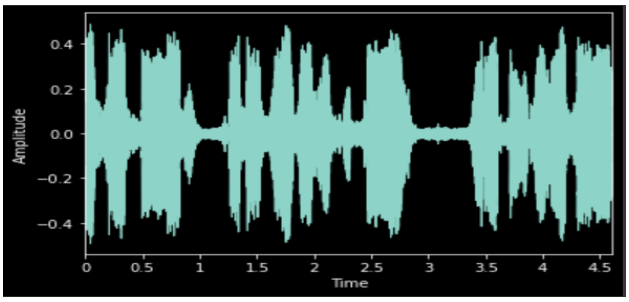
Pre-processing & Pipeline

Audio files are analog in nature which have smooth curves and continuity, analog signals are flagged at specific points (sampling) to perform ADC (Analog to Digital Conversion) wherein the signal is changed to a multilevel digital signal which represents the signal as a sequence of discrete values. The most common method to perform ADC is PCM (Pulse Code Modulation).

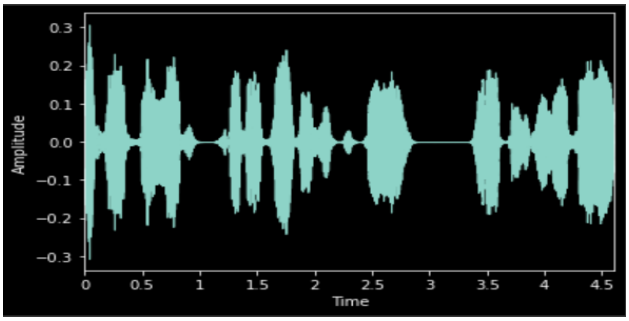
The files have variable duration and have 82000 sample rate which has been reduced to 16000 sample rate and 16 bit-depth.

This is a digitalised audio signal of a male. Noise in the signal should be removed using denoising techniques which includes calculating audio statistics.

Pure signal – Time domain



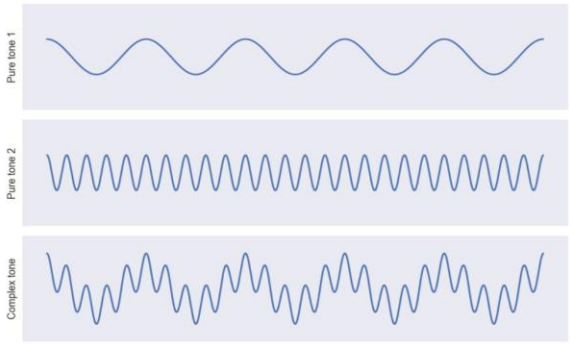
Denoised signal - Time domain



Audio signal can be interpreted either in time domain as shown above, frequency domain or spectral domain. A time domain graph shows how a signal changes over time , whereas frequency domain shows how much of the signals lies within each frequency band over a range of frequencies.

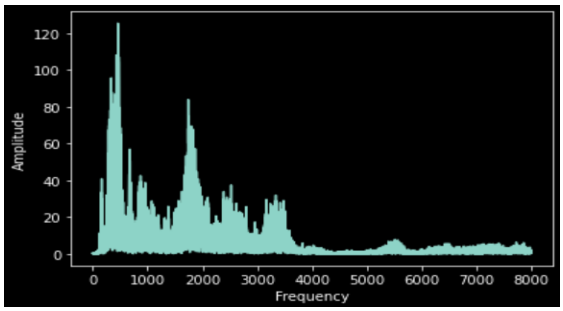
A time domain graph can be converted to frequency domain using Discrete Fourier Transform (DFT)/ Fast

Fourier Transform (FFT) which decomposes the main signal into multiple sinusoids.

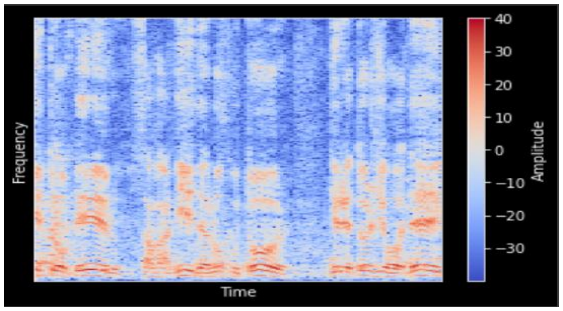


A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time, also called sonographs or voiceprints. Spectrogram is derived using a shortened version of Fourier Transform i.e., Short Time Fourier Transform (STFT).

Frequency domain



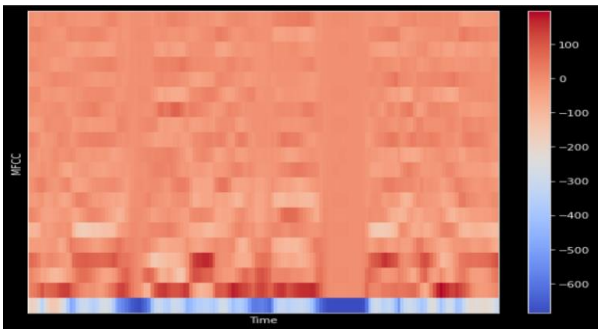
Spectrogram



Spectral plots display three properties of audio signals on a single graph, time, frequency, and colors displaying the amplitude (in dB). Brighter regions denote higher intensity over a range of frequency and time interval. In comparison to an FFT, a spectrogram gives a better look into how the vibration changes over time.

Similar to feature extraction in Convolution Neural Network, where output of a convolution layer gives a feature map with prominent edges of an input image, feature extraction in audio signals is done using *MFCCs* (Mel Frequency Cepstral Coefficients) which are discrete numbers representing a spectrum. MFCC coefficients contain information about the rate changes in the different spectrum bands.

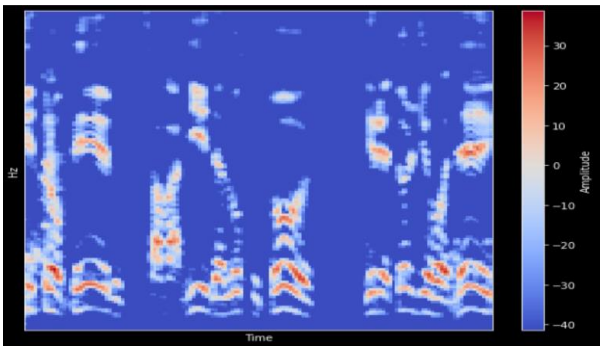
MFCCs



Positive cepstral coefficients denote most of spectral energy is concentrated in low frequency regions, and negative values denote higher density of spectral energy in high frequency regions.

These MFCCs are then fed to a Neural Network to classify class labels, genders in this case.

Mel spectrogram



The spectrogram images can be passed to CNN to classify gender labels.

Literature Review

The task is that of a binary classification based on speech signals. Two different kinds of model can be implemented, a CNN model or a Fully connected Neural Network. The first uses spectrograms generated from Fourier transforms and the second uses MFCCs (audio feature extracted from multiple short Fourier transforms). In either case the final layer is a single neuron that outputs 0/1. 'librosa' Python module is used for audio handling.

CNN

The generated Mel spectrograms are passed are used as images for an input in CNN, techniques used can be either be a Vanilla CNN or incorporating transfer learning and using pretrained models such as

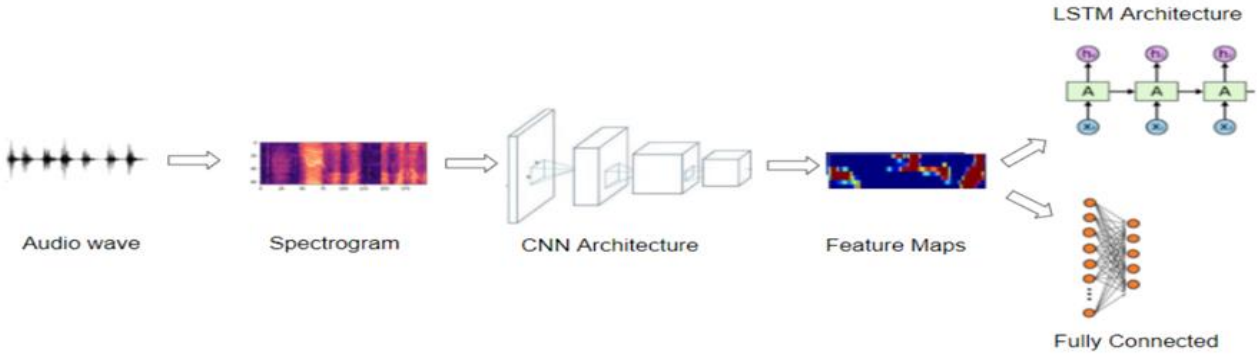
ResNet50, VGG-16 or Inception network trained on ImageNet and fine tune the parameters we want to keep. Even though the pretrained models are trained on a completely different set of images, the optimized parameters help, and we can decide what layers to freeze and what to cull.

Spectrograms are an equivalent compact representation of an audio signal, somewhat like a 'fingerprint' of the signal. It is an elegant way to capture the essential features of audio data as an image.

Start with raw audio data and pre-process it and convert the audio data into its corresponding spectrograms.

You can use pre-processing techniques to augment the spectrogram image, and now that we have image data use the techniques mentioned in the beginning for classification. For instance, for an audio classification problem such as this, you would pass this through some fully connected linear layers in the end.

This is also tried with a novel model which has multiple pooling and dropout layers, as well as a LSTM network where samples of trailing signal help in identifying different pitch and inflections in speeches of people

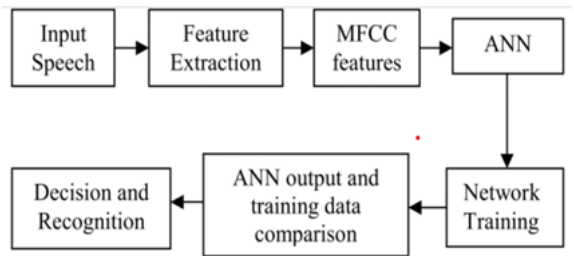


Vanilla Neural Network

The Cepstral coefficients extracted during EDA can also be passed to a Vanilla neural network. Feed forward network with back propagation algorithm is used. For feature extraction of speech MFCCs are used which give us a set of feature vectors of speech waveform. Various research has shown MFCC to be more accurate and effective than other feature extraction techniques in speech recognition.

This may be attributed because MFCCs models the human auditory perception with regard to frequencies, which then can represent sound better. This technique is often used to create the fingerprint of the sound files.

Pipeline for ANN



Using 'librosa' python module MFCCs are extracted. There are 20 coefficients over 145 different time frames. These numbers act as feature vectors to input layers of an ANN.

There is an imbalance of entries, so under sampling is performed to include about 1000 files of each class to avoid the network from learning any class 'too much' of one class.

Comparison

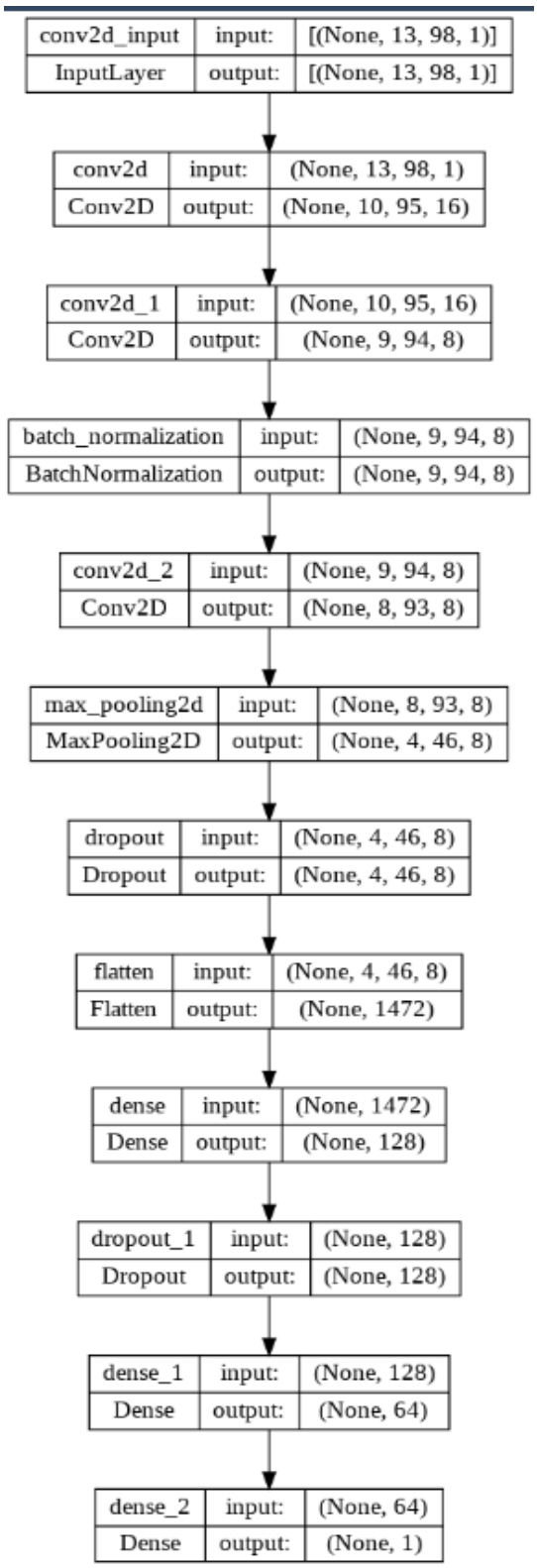
The only drawback of a Vanilla network is the number of parameters it must work with. It becomes exceedingly expensive, compute wise. On the contrary, CNN with its properties as sparse connectivity and shared weights helps reduce the total weights and training time, but here we must plot Mel spectrograms and save these plots as images into a directory to work with. Both have their pros and cons. Final comparison including performance and training time will tabulated in the python notebook.

Conclusion

The vanilla CNN performs relatively well with recall of 60% and the pretrained network (VGG16) trained on ImageNet, that is fed spectrogram images performs very well with classification accuracy of 83%

CNN architecture for MFCC features. This is a straight-forward architecture. This uses SGD along with momentum and decay. The ROC curve provides the optimal threshold for classifying the target label, in this case it is 0.497.

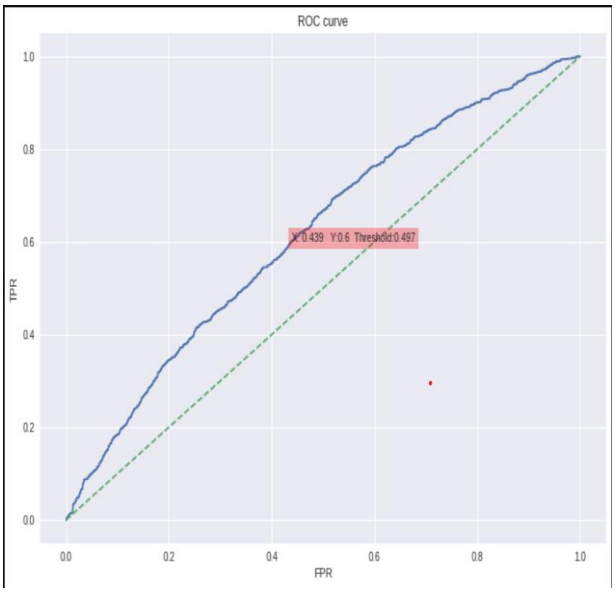
Vanilla CNN architecture



VGG16 for spectrogram images. VGG16 here is pre-trained on ImageNet dataset and has pretty good accuracy for male/female classification. SGD is used for optimizing the loss function (binary cross-entropy) along with Nesterov momentum.

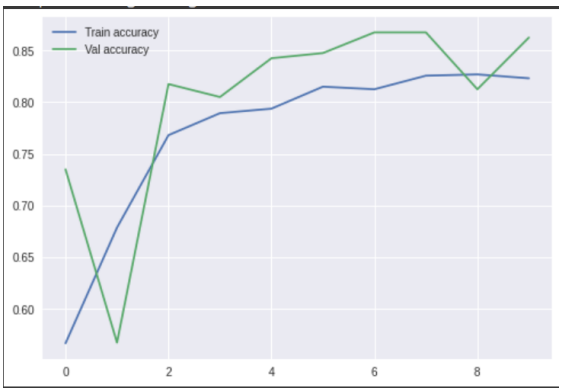
Model	Train accuracy	Validation accuracy
Vanilla CNN	55%	58%
VGG16	82%	86%

ROC Curve for CNN

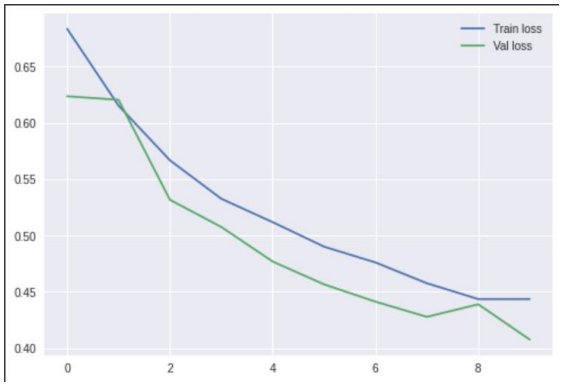


Threshold:0.497, X(FPR): 0.439, Y(Recall): 0.60

VGG accuracy-10 epochs



VGG loss-10 epochs



Final train loss: 0.4375

Final validation loss: 0.4355

Hyperparameter tuning

All the hyperparameters for vanilla CNN are tuned using Keras Tuner, which return the best possible parameters for maximum accuracy.

In case of VGG16, it is already pretrained on a huge dataset so the calculated weights for lower layers perform very well for spectrogram images.

References

https://ieeexplore.ieee.org/document/6850680

https://www.semanticscholar.org/paper/Speech-Recognition-using-MFCC-and-Neural-Networks-Srivastava/da2852be61c71379d0d5ef4e9bfb65a659712b96

https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5

https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

https://towardsdatascience.com/learning-from-audio-the-mel-scale-mel-spectrograms-and-mel-frequency-cepstral-coefficients-f5752b6324a8