Speech Emotion Recognition

Objective

The objective is to make a machine learning model which can classify speech into various categories of emotions.

The application of the speech emotion recognition system include the psychiatric diagnosis, intelligent toys, lie detection, in the call centre conversations which is the most important application for the automated recognition of emotions from the speech, in car board system where information of the mental state of the driver may provide to the system to start his/her safety

Common Technique

Audio files in .wav, .mp3 format

- Pitch
- Speech Energy
- MFCC
- Mel

Simple classification algorithms:

- SVM
- KNN
- Random forest
- Gradient Boosting
- Extra trees

Using Deep learning:

- CNN
- CNN+LSTM

The different emotion to which the audio files belong.

Speech Corpus

Ryerson Speech database has been used. This dataset contains 1440 files(60 trails per actor x 24 actors), total size = 215 MB classified into 8 different classes, namely angry, calm, disgust, happy, fearful, sad, surprised and neutral. The first 7 of them contains 192 files, whereas neutral contains 96 files in .wav format. Each file has a unique filename made up of 7-part numerical identifiers (e.g., 02-01-06-01-02-01-12.wav).:

Filename Identifiers:

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
- Vocal channel (01 = speech,ach 02 = song).
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
- Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
- Repetition (01 = 1st repetition, 02 = 2nd repetition).
- Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

Feature Extraction

‘pyAudioAnalysis’ is used to extract audio features, train and apply audio classifiers, segment an audio stream using supervised or unsupervised methodologies and visualize content relationships. The total number of features implemented in pyAudioAnalysis is 34 which includes Zero Crossing Rate, Energy, Entropy of Energy, Spectral Flux, MFCCs, Chroma Vector etc. The library is written in Python, which is a high-level programming language that has been attracting increasing interest, especially in the academic and scientific community during the past few years

Classifiers Used

Classifier Type	Accuracy
SVM	51%
KNN	55%
Gradient Boosting	47%
Random Forest	52%
Extra Trees	54%

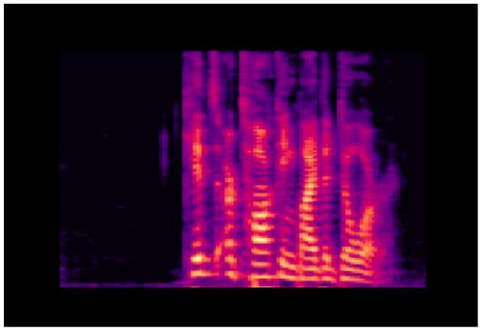
USING DEEP LEARNING

Preprocessing

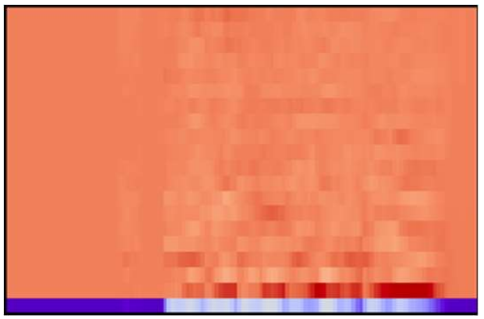
Initially, all audio files are cropped to same length.

Mel and mfcc spectrograms are generated using the functions implemented in the librosa library of python.

These spectrograms generated are used as an input for the CNN.



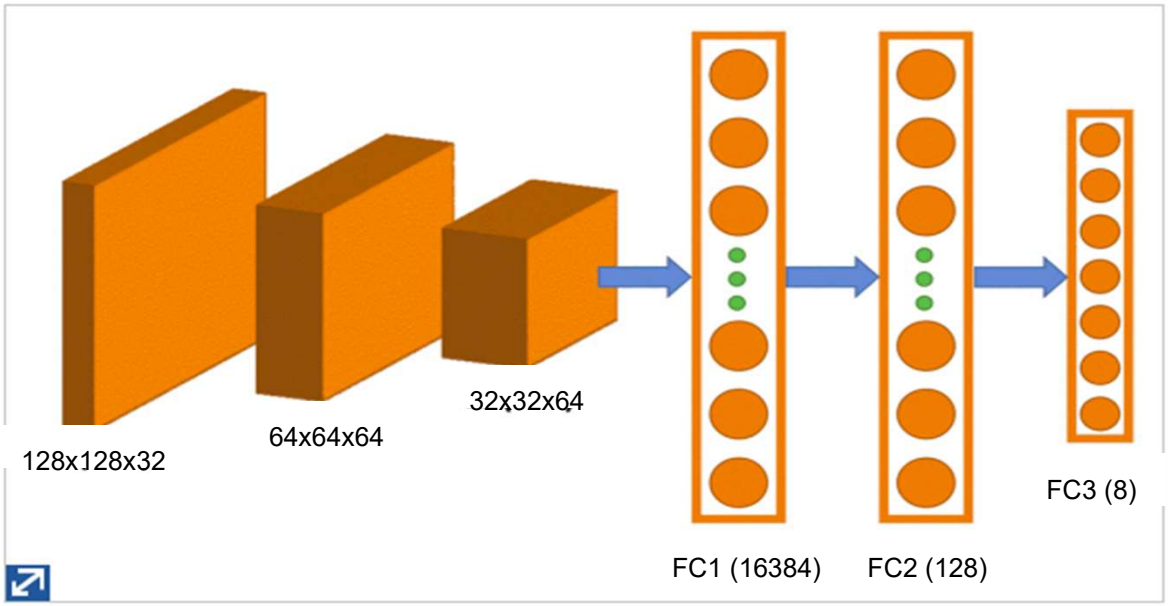
Mel Spectrogram



MFCC Spectrogram

Architecture

The CNN model consist of three convolutional layers(with maxpooling), two fully connected layer and a softmax layer. The input of the network is 128x128. Non-linear activation function such as ReLU is used. In order to avoid overfitting the fully connected layers are followed by dropout layers.



Result:

The resultant network was trained on RYERSON data using ‘adam’ optimiser and ‘categorial crossentropy’ as a loss function.

When the network was trained on 50 epochs, we achieved an accuracy of about **57%.**

Data Augmentation

Why?

The data available in the speech corpus was not sufficient enough to train the deep learning model. So we decided to augment the data so that the size of the database can be increased.

The data augmentation was done by varying the following parameters:

- Pitch - The pitch is increased or decreased by few semitones.
- Speed - The audio is slightly stretched or compressed

It helped us to generate a total of 15840 files belonging to 8 different classes.

Training and Result

The augmented data along with the original data is splitted into 75:25 train, test ratio and trained on the same CNN model for 100 epochs.

It achieved an accuracy of about **86%**

CNN+LSTM

The two fully connected layers in the CNN model are replaced by two LSTM layers comprising of 50 and 20 nodes respectively.

The model achieved an accuracy of **92%.**

Conclusion

Use of CNN-LSTM model for recognition of emotion from speech is a effective step towards designing a generic emotion recognition system. Although the size of data set is not so large the performance of our proposed model is promising enough. Some normalized input data or use of Bidirectional LSTM instead of LSTM can lead to us more better solution. Also the training will produce more convenient outcome if we can feed a larger set of data to the system.

ENJOY DEEP LEARNING