# TABLE OF CONTENTS

## Definitions, Acronyms and Abbreviations

- **MFCC** - Mel-frequency cepstral coefficients are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum")
- **RNN** - Recurrent Neural Networks - a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence
- **CNN** - Convolutional Neural Networks - regularized versions of multilayer perceptrons
- **SVM** - Support Vector Machines - are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis
- **NB** - Naive Bayes - Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other
- **Random Forests** - an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees
- **RAVDESS** - The Ryerson Audio-Visual Database of Emotional Speech and Song is a dynamic, multimodal set of facial and vocal expressions in North American English

## References


- https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0196391
  - Title: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English
  - Received: July 31, 2017
  - Accepted: April 12, 2018
  - Published: May 16, 2018
  - Authors: Steven R. Livingstone, Frank A. Russo
- http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/logan00mel.pdf
  - Title: Mel Frequency Cepstral Coefficients for Music Modeling
  - Published: November 2000
  - Author: Beth Logan
- Kim, Samuel, Panayiotis G. Georgiou, Sungbok Lee, and Shrikanth Narayanan. "Real-time emotion detection system using speech: Multi-modal fusion of different timescale features." In *2007 IEEE 9th Workshop on Multimedia Signal Processing*, pp. 48-51. IEEE, 2007.
- Neiberg, Daniel, Kjell Elenius, and Kornel Laskowski. "Emotion recognition in spontaneous speech using GMMs." In *Ninth International Conference on Spoken Language Processing*. 2006.
- Yu, Feng, Eric Chang, Ying-Qing Xu, and Heung-Yeung Shum. "Emotion detection from speech to enrich multimedia content." In *Pacific-Rim Conference on Multimedia*, pp. 550-557. Springer, Berlin, Heidelberg, 2001.
- Liscombe, Jackson, Jennifer Venditti, and Julia Hirschberg. "Classifying subject ratings of emotional speech using acoustic features." In *Eighth European Conference on Speech Communication and Technology*. 2003.

# 1.0 Introduction

## *1.1 Overview*

Gender/Emotion extraction and classification from Audio signals using RAVDESS Dataset via various Machine Learning and Deep Learning techniques.

## *1.2 Scope*

This project aims to split the audio analysis into three phases:

1) Gender classification
2) Emotion detection
3) Person identification

The main focus was on the first two phases, after which, if the data was extensive enough, the analysis could be extended to the third phase as well, as this problem could be an integral part of certain NLG based AI systems which can provide personalized conversations with respect to the person as well as the emotion that s/he is expressing.

The project does not reach out to Speech Generation, which is considered as a part of future works.

## *1.3 Objective*

Classification problem solved over two phases:

- Gender Classification
  - Models Employed -
    - Support Vector Machines
    - Naive Bayes
    - Artificial Neural Networks
- Emotion Detection
  - Models Employed -
    - Convolutional Neural Networks
    - Long Short Term Memory (LSTM)
    - Random Forests
  - Emotions -
    - Neutral
    - Calm
    - Happy
    - Sad
    - Angry
    - Fearful
    - Surprise
    - Disgust

# 2.0 Literature Survey

These are the few research papers that have worked around the same concept or objective

- **Zhang, B., Essl, G. and Provost, E.M., 2015, September. Recognizing emotion from singing and speaking using shared models. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)* (pp. 139-145). IEEE:** This publication adopted the directed acyclic graph SVM (DAGSVM) as their single-task multi-class emotion classifier. DAGSVM is identical to the one-against-one SVM in the training phase. It constructs a binary classifier for each pair of classes, thus for a multi-class problem with k classes, k(k − 1)/2 classifiers are trained.

- **Schuller, Björn, et al. "Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge." Speech Communication 53.9-10 (2011): 1062-1087. :** Some complex asynchronous Hidden Markov Models (HMM), Dynamic Bayesian Networks (DBN), general Graphical Models or Multidimensional Dynamic Time Warp (DTW) and Meta-Classification were created to solve the problem.

- **Gao, Yuanbo, et al. "Speech emotion recognition using local and global features." *International Conference on Brain Informatics*. Springer, Cham, 2017.** : Spectral features including pitch, MFCC, intensity, ZCR and LSP were used to establish the emotion recognition model with SVM classifier. In particular, they found different frame durations to have different influences on final results. So, Depth-First-Search method is applied to find the best parameters.

- **Livingstone, Steven R., Katlyn Peck, and Frank A. Russo. "Ravdess: The ryerson audio-visual database of emotional speech and song." Annual meeting of the canadian society for brain, behaviour and cognitive science. 2012.**

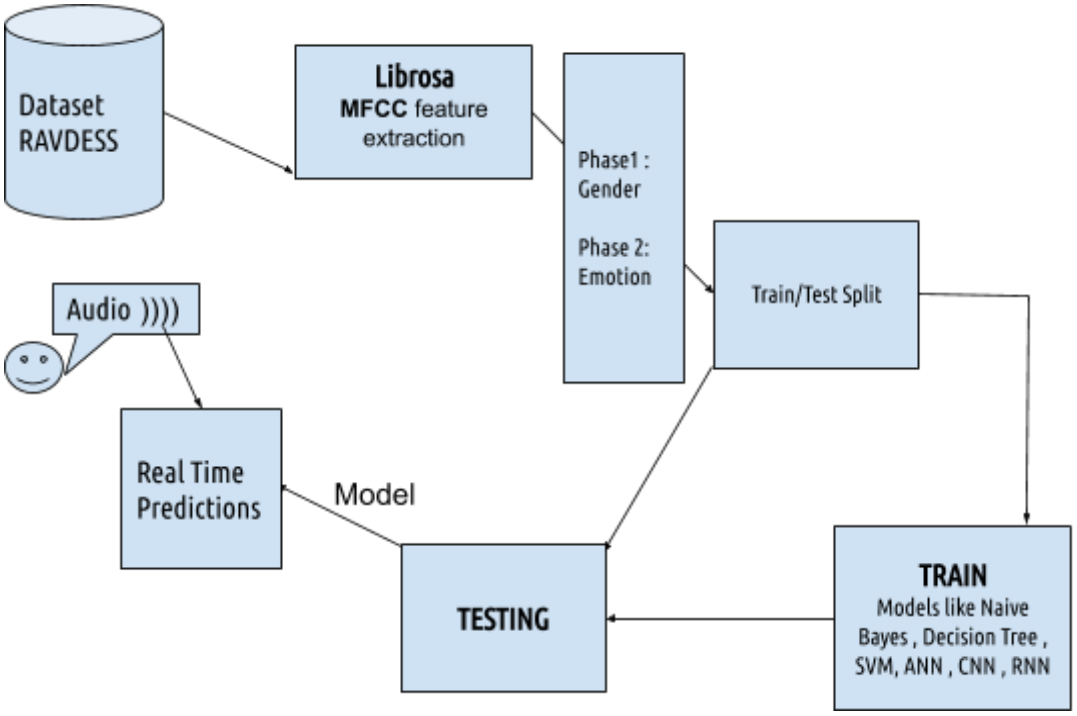
- **He, Zhihao, et al. "Human emotion recognition in video using subtraction pre-processing." 11th International Conference on Machine Learning and Computing. 2019.**

# 3.0 Methodology

## *3.1 Proposed Approach*

The proposed approach was to identify and extract the key features from audio clips in the form of Mel-Frequency Cepstral Coefficients. These features were then averaged out for the entire audio clip to bring down the amount of data (and thus, the computational power needed), and labelled according to gender as well as emotions. Once the data was obtained, various machine learning models were deployed on it and their accuracies were compared.

### *3.2 High Level System Architecture*




## 4.0 Environment Requirements

### *4.1 Hardware Requirements*

The entire project was developed on the cloud using Google Colab, a GPU supported research tool for Machine Learning.

- Version: 1
- Source - Google

### *4.2 Software Requirements*

Python - an interpreted, high-level, general-purpose programming language.

- Version: 3.6
- Source - Python Software Foundation (https://www.python.org/)

Librosa - a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

- Version: 0.6.3
- Source - https://librosa.github.io/librosa/

Tensorflow - an end-to-end open source platform for machine learning.

- Version: 1.13
- Source - https://www.tensorflow.org/api_docs/python/tf

Keras - an open-source neural-network library written in Python, which runs on top of Tensorflow. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

- Version: 2.2.4
- Source - https://keras.io/

Pandas - a software library written for the Python programming language for data manipulation and analysis

- Version: 0.24.1
- Source - https://pandas.pydata.org/

Numpy - a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

- Version: 1.16.2
- Source - https://www.numpy.org/

Sklearn - a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

- Version: 0.20.2
- Source - https://scikit-learn.org/stable/

### *4.3 Data Requirements*

The dataset used is called **RAVDESS** (**Ryerson Audio-Visual Database of Emotional Speech and Song**). The complete version of this dataset includes vocal expressions and facial expressions in North American English. Since our goal is to analyse audio, a subset of this dataset (audio only) has been used.

The complete Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains 7356 files (total size: 24.8 GB). The database contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

**Portion of Dataset used :**

Speech file (Audio_Speech_Actors_01-24.zip, 215 MB) contains 1440 files: 60 trials per actor x 24 actors = 1440.

**Link for the dataset:** https://smartlaboratory.org/ravdess/

## 5.0 Proposed Approach

To examine the various machine learning approaches we first looked at the layman's understanding of sound. ie : Loudness(Amplitude) + Pitch(Frequency). These are simple attributes which generally clash for various sounds.

Now on a perceptual scale (ie: human listeners distinguishing the sounds) different sounds can be distinguished easily.

**MEL Scale :** It is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The reference point between this scale and normal frequency measurement is defined by assigning a perceptual pitch of 1000 mels to a 1000 Hz tone, 40 dB above the listener's threshold.

So the first phase in our pipeline for preprocessing was to convert the audio signal to features in the mel scale. ie :

1. **Take the Fourier transform of (a windowed excerpt of ) a signal.**
2. **Map the powers of the spectrum obtained above onto the mel scale**

Once we have the filterbank energies, the next step is to

3. **Take the logarithm of the filters**. This is also motivated by human hearing: we don't hear loudness on a linear scale. Generally to double the perceived volume of a sound we need to put 8 times as much energy into it.

The final step is to

4. **Compute the DCT:Discrete Cosine Transform of the log filterbank energies.** There are 2 main reasons this is performed

Because our filterbanks are all overlapping, the filterbank energies are quite correlated with each other. The DCT decorrelates the energies which means diagonal covariance matrices can be used to model the features in e.g. a HMM classifier.

The four steps given above provide us with what we know as **MFCC features.** These features were extracted with various sampling rates and trained with different machine learning models and deep learning networks.

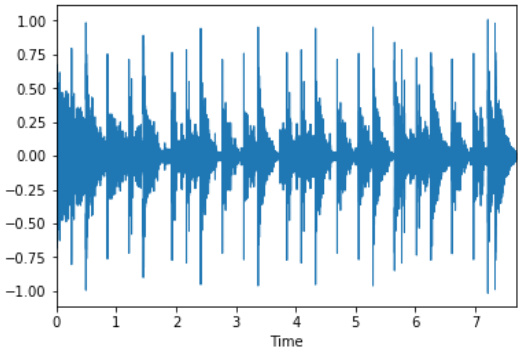



Figure : Audio signal visualization

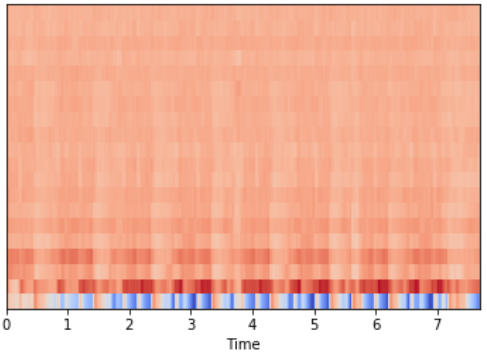



Figure : MFCC visualization

## 6.0 Results

Many models and variations of neural networks were tried and the best models were picked after testing on unseen data points. We keep in mind that the data chosen was limited and not very extensive to learn a lot of information without overfitting such that it generalizes well with any new audio file.

Also the computational constraints prevented us from training complex models.

Gender Classification: ( Binary Classification )

| Machine Model Used | Accuracy |
|---|---|
| Naive Bayes | 98% |
| Support Vector Machine | 94% |
| Artificial Neural Network | 95.36% |

Emotion Classification: ( 8 classes )

| Machine Model Used | Accuracy |
|---|---|
| CNN | 64.9% |
| LSTM | 55.74% |
| Random Forest | 51% |

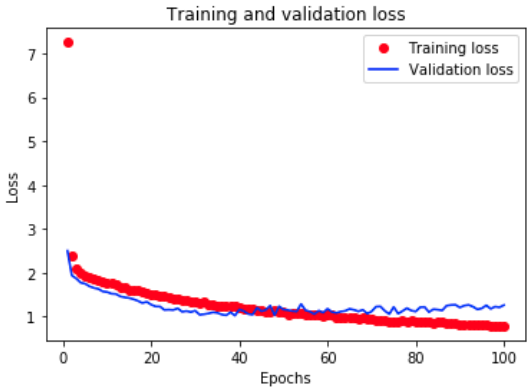
# Training

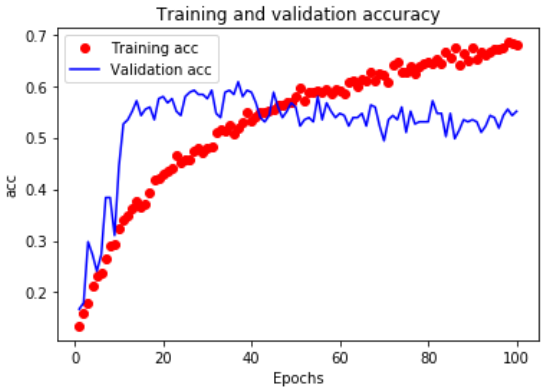
```
Epoch 129/150
921/921 [==============================] - 0s 467us/step - loss: 0.0293 - acc: 0.9913 - val_loss: 3.1144 - val_acc: 0.5801
Epoch 130/150
921/921 [==============================] - 0s 473us/step - loss: 0.0206 - acc: 0.9902 - val_loss: 3.1101 - val_acc: 0.5541
Epoch 131/150
921/921 [==============================] - 0s 469us/step - loss: 0.0327 - acc: 0.9924 - val_loss: 2.8661 - val_acc: 0.5758
Epoch 132/150
921/921 [==============================] - 0s 474us/step - loss: 0.0312 - acc: 0.9891 - val_loss: 3.1370 - val_acc: 0.5281
Epoch 133/150
921/921 [==============================] - 0s 465us/step - loss: 0.0311 - acc: 0.9891 - val_loss: 3.1298 - val_acc: 0.5801
Epoch 134/150
921/921 [==============================] - 0s 462us/step - loss: 0.0243 - acc: 0.9946 - val_loss: 3.0585 - val_acc: 0.5628
Epoch 135/150
921/921 [==============================] - 0s 462us/step - loss: 0.0267 - acc: 0.9935 - val_loss: 3.2419 - val_acc: 0.5801
Epoch 136/150
921/921 [==============================] - 0s 455us/step - loss: 0.0268 - acc: 0.9870 - val_loss: 3.1304 - val_acc: 0.5498
Epoch 137/150
921/921 [==============================] - 0s 470us/step - loss: 0.0313 - acc: 0.9902 - val_loss: 3.0026 - val_acc: 0.5281
Epoch 138/150
921/921 [==============================] - 0s 469us/step - loss: 0.0267 - acc: 0.9913 - val_loss: 3.0645 - val_acc: 0.5281
Epoch 139/150
921/921 [==============================] - 0s 462us/step - loss: 0.0274 - acc: 0.9913 - val_loss: 3.2295 - val_acc: 0.5411
Epoch 140/150
921/921 [==============================] - 0s 464us/step - loss: 0.0304 - acc: 0.9902 - val_loss: 3.3193 - val_acc: 0.5671
Epoch 141/150
921/921 [==============================] - 0s 466us/step - loss: 0.0263 - acc: 0.9902 - val_loss: 3.0177 - val_acc: 0.5758
Epoch 142/150
921/921 [==============================] - 0s 474us/step - loss: 0.0480 - acc: 0.9805 - val_loss: 3.0049 - val_acc: 0.5671
Epoch 143/150
921/921 [==============================] - 0s 466us/step - loss: 0.0442 - acc: 0.9881 - val_loss: 3.0273 - val_acc: 0.5714
Epoch 144/150
921/921 [==============================] - 0s 468us/step - loss: 0.0320 - acc: 0.9881 - val_loss: 3.0827 - val_acc: 0.5758
Epoch 145/150
921/921 [==============================] - 0s 478us/step - loss: 0.0205 - acc: 0.9924 - val_loss: 3.2681 - val_acc: 0.5758
Epoch 146/150
921/921 [==============================] - 0s 484us/step - loss: 0.0196 - acc: 0.9902 - val_loss: 2.9830 - val_acc: 0.5801
Epoch 147/150
921/921 [==============================] - 0s 465us/step - loss: 0.0249 - acc: 0.9913 - val_loss: 3.0820 - val_acc: 0.5887
Epoch 148/150
921/921 [==============================] - 0s 469us/step - loss: 0.0148 - acc: 0.9957 - val_loss: 3.1727 - val_acc: 0.5758
Epoch 149/150
921/921 [==============================] - 0s 485us/step - loss: 0.0271 - acc: 0.9924 - val_loss: 3.2528 - val_acc: 0.5714
Epoch 150/150
921/921 [==============================] - 0s 466us/step - loss: 0.0231 - acc: 0.9913 - val_loss: 3.3348 - val_acc: 0.5758
```

# CNN Architecture

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv1d_17 (Conv1D) | (None, 48, 128) | 512 |
| dropout_6 (Dropout) | (None, 48, 128) | 0 |
| conv1d_18 (Conv1D) | (None, 46, 64) | 24640 |
| dropout_7 (Dropout) | (None, 46, 64) | 0 |
| flatten_7 (Flatten) | (None, 2944) | 0 |
| dense_27 (Dense) | (None, 8) | 23560 |

Total params: 48,712
Trainable params: 48,712
Non-trainable params: 0

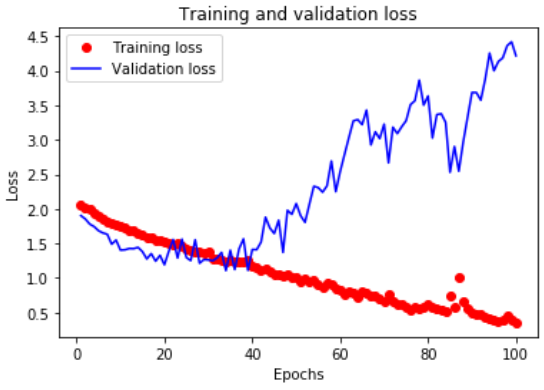


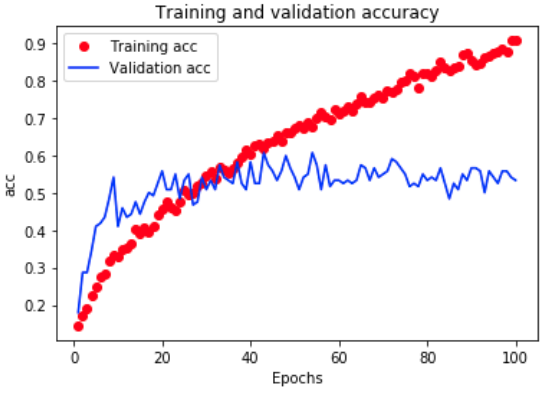




# LSTM Architecture

| Layer (type) | Output Shape | Param # |
|---|---|---|
| lstm_2 (LSTM) | (None, 128) | 66560 |
| dense_4 (Dense) | (None, 64) | 8256 |
| dropout_3 (Dropout) | (None, 64) | 0 |
| activation_4 (Activation) | (None, 64) | 0 |
| dense_5 (Dense) | (None, 32) | 2080 |
| dropout_4 (Dropout) | (None, 32) | 0 |
| activation_5 (Activation) | (None, 32) | 0 |
| dense_6 (Dense) | (None, 8) | 264 |
| activation_6 (Activation) | (None, 8) | 0 |

Total params: 77,160
Trainable params: 77,160
Non-trainable params: 0



## 7.0 Conclusions

This project enhanced the use of various machine learning techniques and combining various audio signal processing concepts. It bridges the gap between audio waves and recognition of patterns and information (like emotions) from them in a learnable form which can be applied to various applications that humans can easily perceive.

## 8.0 Future Work

Analyzing other emotions commonly encountered in day to day lives would help in the study of the topic. This project could also be combined with Facial Expressions to enhance the number of features obtained, which might provide us with better results. An extension to Speech/Response Generation based on emotion would pave the way to an application for NLG based AI bots.

## 9.0 References


- https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0196391
  - Title: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English
  - Received: July 31, 2017
  - Accepted: April 12, 2018
  - Published: May 16, 2018
  - Authors: Steven R. Livingstone, Frank A. Russo
- http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/logan00mel.pdf
  - Title: Mel Frequency Cepstral Coefficients for Music Modeling
  - Published: November 2000
  - Author: Beth Logan
- Kim, Samuel, Panayiotis G. Georgiou, Sungbok Lee, and Shrikanth Narayanan. "Real-time emotion detection system using speech: Multi-modal fusion of different timescale features." In *2007 IEEE 9th Workshop on Multimedia Signal Processing*, pp. 48-51. IEEE, 2007.
- Neiberg, Daniel, Kjell Elenius, and Kornel Laskowski. "Emotion recognition in spontaneous speech using GMMs." In *Ninth International Conference on Spoken Language Processing*. 2006.
- Yu, Feng, Eric Chang, Ying-Qing Xu, and Heung-Yeung Shum. "Emotion detection from speech to enrich multimedia content." In *Pacific-Rim Conference on Multimedia*, pp. 550-557. Springer, Berlin, Heidelberg, 2001.
- Liscombe, Jackson, Jennifer Venditti, and Julia Hirschberg. "Classifying subject ratings of emotional speech using acoustic features." In *Eighth European Conference on Speech Communication and Technology*. 2003.