SPEECH ACCENT ARCHIVE

Submitted by team-05

Alekhya Thangallapelly (12567125) (Class ID: 1)

Tejaswini Rayapati (16292187) (Class ID: 23)

Srujan Reddy Tirupally (16291102) (Class ID:21)

Saishivareddy Vootukuru(16292488)(Class ID:16)

Member Name	Email	Class ID	Contribution
Alekhya Thangallapelly	atzd3@mail.umkc.edu	01	convolutional neural network and Audio feature(Audio wave)
Tejaswini Rayapati	trk6f@mail.umkc.edu	23	k-nearest neighbors model,Audio features(chroma, Log Mel-spectogram)
Srujan Reddy Tirupally	stn3c@mail.umkc.edu	21	convolutional neural network and Audio(Mel Frequency Cepstral Coefficient)
Saishivareddy Vootukuru	trk6f@mail.umkc.edus	16	k-nearest neighbors model and Harmonic-percussive source separation

Technologies Used:

Google Colab

Pycharm

Source:

The dataset has taken from –

https://www.kaggle.com/rtatman/speech-accent-archive.

Inspiration:
Motivation for the following data set had been originated from the below mentioned reasons. Any individual who sees outside complement as fascinating.
Engineers who train speech recognition machines. ESL instructors who teach non-local speakers of English.

Approach:
Changed over wav audio records into Mel Frequency Cepstral Coefficients graph. The MFCC was taken care of into a 2-Dimensional Convolutional Neural Network (CNN) to anticipate the local language class. • ├── src ├── accuracy.py ├── fromwebsite.py ├── getaudio.py ├── getsplit.py ├── trainmodel.py ├── models ├── cnn_model138.h5 ├── logs ├── events.out.tfevents.1506987325.ip-172-31-47-225 └── sound. As we are utilizing audio records, it incorporates more information for speed and adaptability we are utilizing cnn model.

Dependencies:
Python 3.5 (https://www.python.org/download/releases/2.7/)

o Keras (https://keras.io/)

Numpy (http://www.numpy.org/)

BeautifulSoup (https://www.crummy.com/software/BeautifulSoup/)

Pydub (https://github.com/jiaaro/pydub)

Sklearn (http://scikit-learn.org/stable/)

Librosa (http://librosa.github.io/librosa/)

Project description:
Each individual has their own lingos or mannerisms where they talk. This project rotates around the background detections of each individual utilizing their discourses. The objective in this project is to arrange different kinds of accents, explicitly outside accents, by the local language of the speaker. This undertaking permits to distinguish the segment and semantic foundations of the speakers by contrasting diverse discourse yields and the discourse highlight chronicle dataset so as to figure out which factors are key indicators of each complement. The discourse complement document shows that accents are methodical as opposed to simply mixed up discourse. Given an account of a speaker talking a known content of English words, this undertaking predicts the speaker's local language.

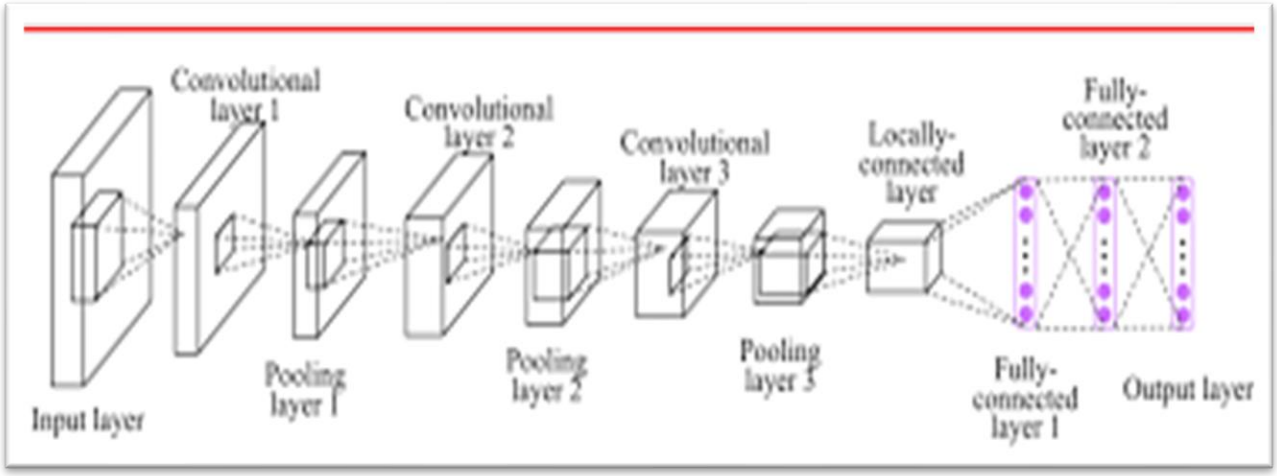
Dataset:
This dataset contains 2140 discourse tests, each from an alternate talker perusing a similar understanding entry. Talkers originate from 177 nations and have 214 distinctive local languages. Every talker is talking in English.

In this project underneath referenced examination will be performed.

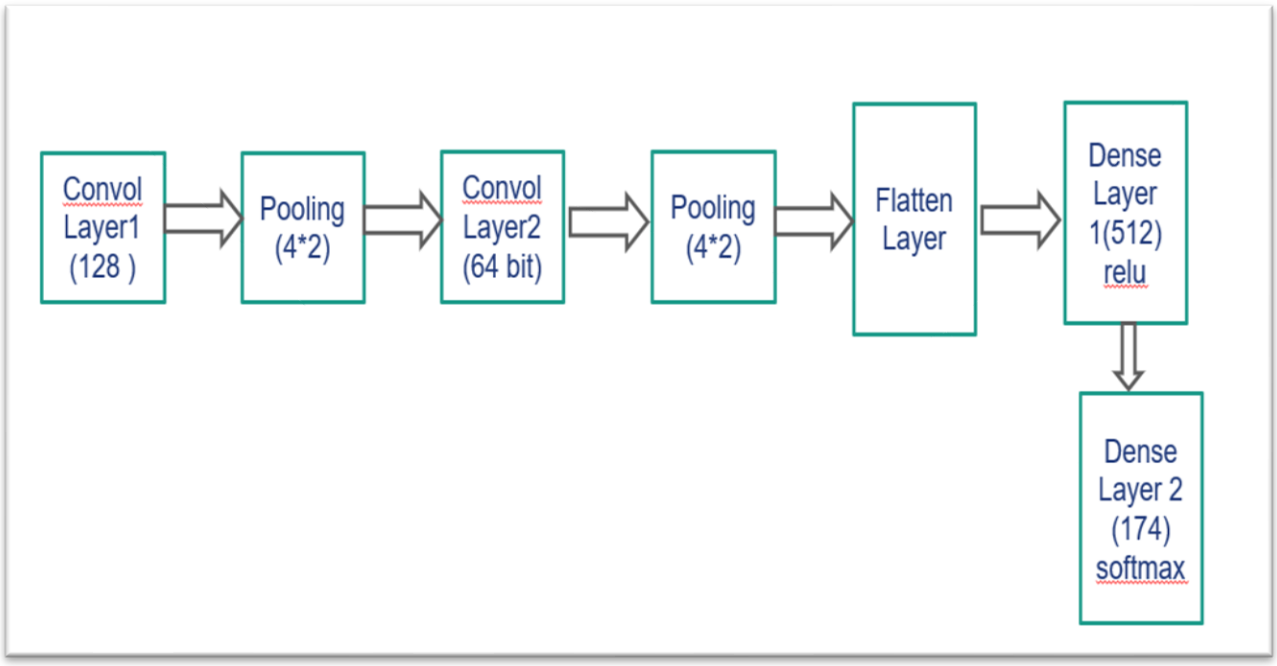
- Audio Sampling at different frequencies
- MFCC (Mel-Frequency Cepstral Coefficient)
- Log Mel-Spectrogram
- Harmonic-percussive source detachment
- Chroma

Data set Description:
Speech Accent Archive dataset contains 2140 speech samples, each from a different talker reading the same reading passage.
Speakers come from 177 countries and have 214 different native languages.
Each talker speaks in English.
Csv file has following labels: age_onset, birthplace, filename(audio), native_language, sex, speakerid, country.

ARCHITECTURE:



Overall Architecture



Audio feature extraction and comparing all the accents.

1.Audio wave:

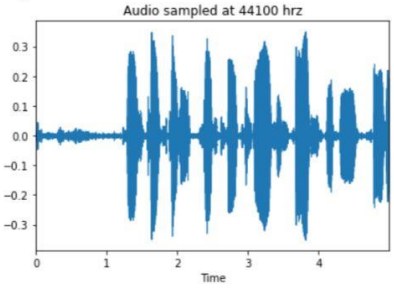
Different sampling rates are plotted, at 44kHz, 6kHz and 1000kHz and perceive how they contrast, utilizing the female rendition of the audio.We have utilized female french voice.

We have seen that a male speaker appears to have an increasingly reliable pitch additional time versus the female partner, whom we can clearly observe a conspicuous enormous spike in amplitude..

```
SAMPLE_RATE = 44100
fname_f = '/content/drive/My Drive/Recordings/recordings/french1.wav'
y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)

plt.figure(figsize=(10, 4))
plt.figure()
librosa.display.waveplot(y, sr=sr)
plt.title('Audio sampled at 44100 hrz')
```

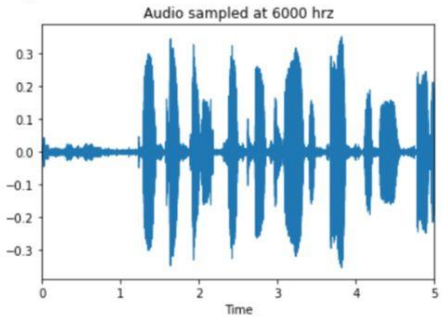
```
Text(0.5, 1.0, 'Audio sampled at 44100 hrz')
<Figure size 720x288 with 0 Axes>
```




```
[ ] SAMPLE_RATE = 6000
    fname_f = '/content/drive/My Drive/Recordings/recordings/french1.wav'
    y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)

    plt.figure(figsize=(10, 4))
    plt.figure()
    librosa.display.waveplot(y, sr=sr)
    plt.title('Audio sampled at 6000 hrz')
```

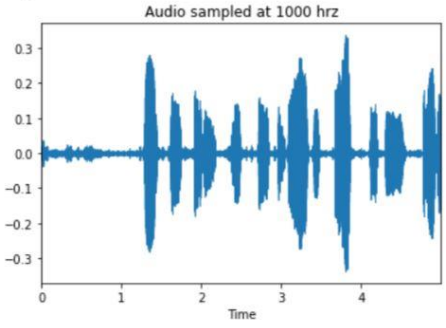
```
Text(0.5, 1.0, 'Audio sampled at 6000 hrz')
<Figure size 720x288 with 0 Axes>
```



```
[ ] SAMPLE_RATE = 1000
    fname_f = '/content/drive/My Drive/Recordings/recordings/french1.wav'
    y, sr = librosa.core.load(fname_f, sr=SAMPLE_RATE, duration = 5)

    plt.figure(figsize=(10, 4))
    plt.figure()
    librosa.display.waveplot(y, sr=sr)
    plt.title('Audio sampled at 1000 hrz')
```

```
Text(0.5, 1.0, 'Audio sampled at 1000 hrz')
<Figure size 720x288 with 0 Axes>
```

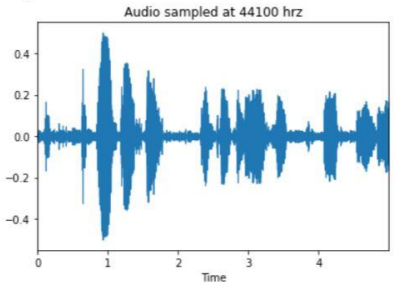


We analyzed male voice at 44kHz, 6kHz and 1000kHz.

```
[ ] SAMPLE_RATE = 44100
    fname_f = '/content/drive/My Drive/Recordings/recordings/french2.wav'
    y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)

    plt.figure(figsize=(10, 4))
    plt.figure()
    librosa.display.waveplot(y, sr=sr)
    plt.title('Audio sampled at 44100 hrz')
```

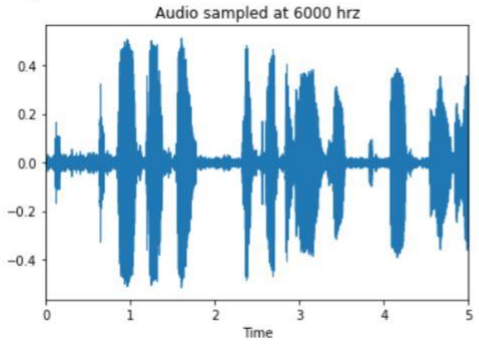
```
Text(0.5, 1.0, 'Audio sampled at 44100 hrz')
<Figure size 720x288 with 0 Axes>
```



```
[ ] SAMPLE_RATE = 6000
    fname_f = '/content/drive/My Drive/Recordings/recordings/french2.wav'
    y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)

    plt.figure(figsize=(10, 4))
    plt.figure()
    librosa.display.waveplot(y, sr=sr)
    plt.title('Audio sampled at 6000 hrz')
```

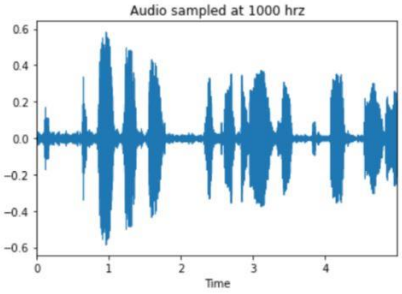
```
Text(0.5, 1.0, 'Audio sampled at 6000 hrz')
<Figure size 720x288 with 0 Axes>
```




```
[ ] SAMPLE_RATE = 1000
    fname_f = '/content/drive/My Drive/Recordings/recordings/french2.wav'
    y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)

    plt.figure(figsize=(10, 4))
    plt.figure()
    librosa.display.waveplot(y, sr=sr)
    plt.title('Audio sampled at 1000 hrz')
```

```
Text(0.5, 1.0, 'Audio sampled at 1000 hrz')
<Figure size 720x288 with 0 Axes>
```



We have observed that a male speaker seems to have a more consistent pitch overtime versus the female counter part, whom we can clearly see an obvious huge spike in amplitude.

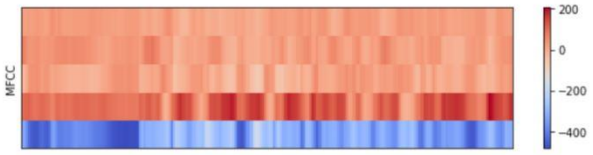
2.MFCC

On the element we have chipped away at the MFCC which is a represetation of the vocal tract that delivers the sound.

```
[ ] SAMPLE_RATE = 22050
    fname_f = '/content/drive/My Drive/Recordings/recordings/french1.wav'
    y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5) # Chop audio at 5 secs...
    mfcc = librosa.feature.mfcc(y=y, sr=SAMPLE_RATE, n_mfcc = 5) # 5 MFCC components

    plt.figure(figsize=(10, 8))
    plt.subplot(3,1,1)
    librosa.display.specshow(mfcc)
    plt.ylabel('MFCC')
    plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7fc2fe6809b0>
```

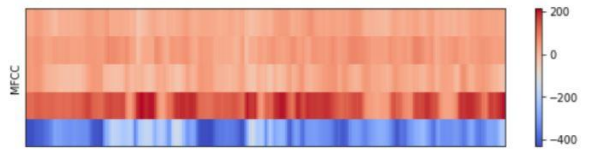


MFCC for male version

```
[ ] SAMPLE_RATE = 22050
    fname_m = '/content/drive/My Drive/Recordings/recordings/french2.wav'
    y, sr = librosa.load(fname_m, sr=SAMPLE_RATE, duration = 5)
    mfcc = librosa.feature.mfcc(y=y, sr=SAMPLE_RATE, n_mfcc = 5)

    plt.figure(figsize=(10, 8))
    plt.subplot(3,1,1)
    librosa.display.specshow(mfcc)
    plt.ylabel('MFCC')
    plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7fc2fe702320>
```



3.Log Mel-Spectogram

A period by frequency representation of the audio wave form is known as a spectogram. That spectogram is then mapped to the Mel-scales subsequently giving us the Mel-spectogram. We prefer log as human impression of sound force is sign in nature.

We have observed that the female version reaches a higher frequency compared to the male counterpart.

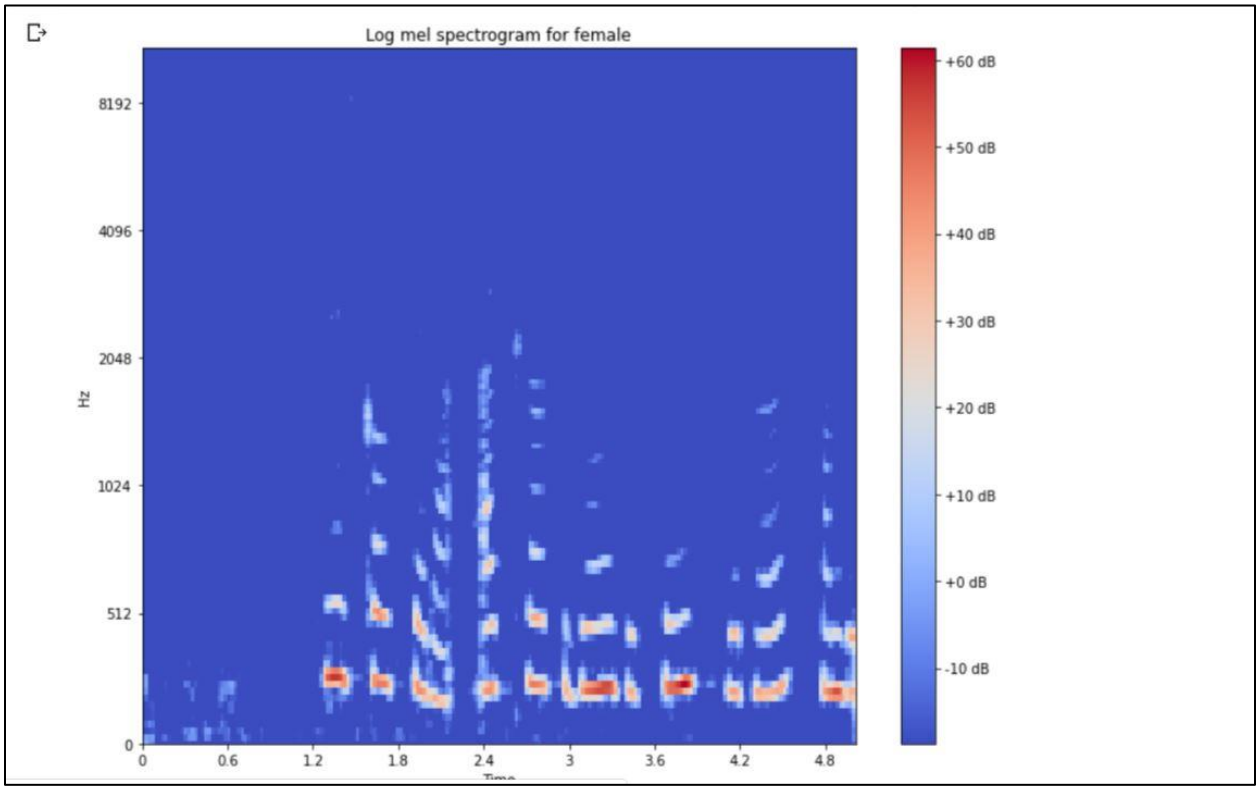
Log Mel-spectogram for Female

```
[ ]  SAMPLE_RATE = 22050
     fname_f = '/content/drive/My Drive/Recordings/recordings/french1.wav'
     y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5) # Chop audio at 5 secs...
     melspec = librosa.feature.melspectrogram(y, sr=sr, n_mels=128)

     # Convert to log scale (dB). We'll use the peak power (max) as reference.
     log_S = librosa.amplitude_to_db(melspec)

     # Display the log mel spectrogram
     plt.figure(figsize=(10,8))
     librosa.display.specshow(log_S, sr=sr, x_axis='time', y_axis='mel')
     plt.title('Log mel spectrogram for female')
     plt.colorbar(format='%+02.0f dB')
     plt.tight_layout()
```

Log Mel-spectogram for Male



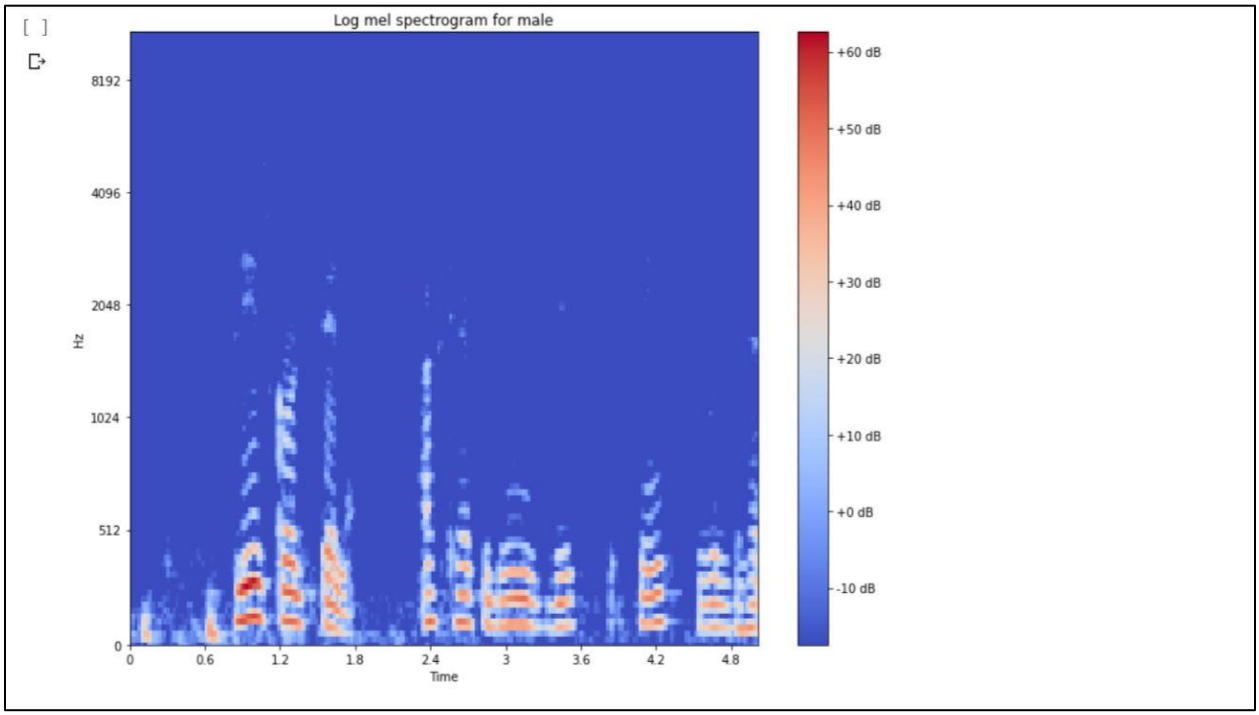
We have observed that the female version reaches a higher frequency compared to the male counterpart.

4.Harmonic-percussive source separaion:
This feature seperates the harmonic and the percussive source of an audio file. percussive means a drum sound or sudden disturbance. As we have the uniformed audio file we have the same output.

```
[ ]  SAMPLE_RATE = 22050
     fname_m = '/content/drive/My Drive/Recordings/recordings/french2.wav'
     y, sr = librosa.load(fname_m, sr=SAMPLE_RATE, duration = 5) # Chop audio at 5 secs...
     melspec = librosa.feature.melspectrogram(y, sr=sr, n_mels=128)

     # Convert to log scale (dB). We'll use the peak power (max) as reference.
     log_S = librosa.amplitude_to_db(melspec)

     # Display the log mel spectrogram
     plt.figure(figsize=(10,8))
     librosa.display.specshow(log_S, sr=sr, x_axis='time', y_axis='mel')
     plt.title('Log mel spectrogram for male')
     plt.colorbar(format='%+02.0f dB')
     plt.tight_layout()
```

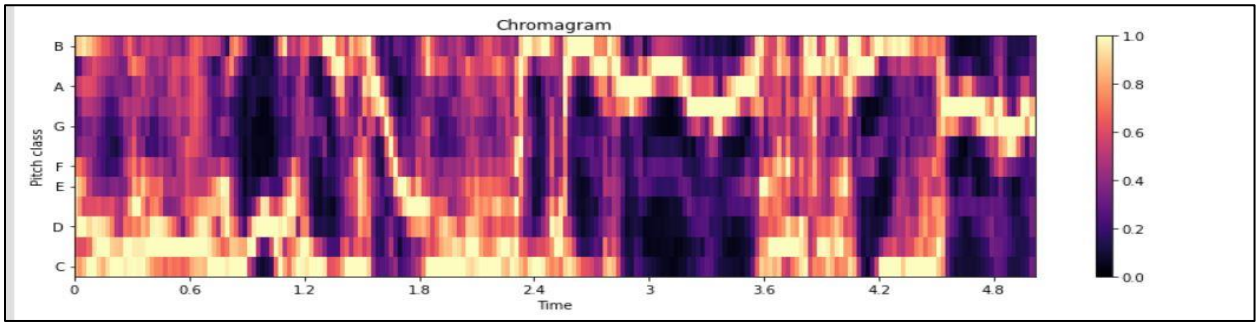


5.CHROMA

Chroma is most discriminative feature between male and female

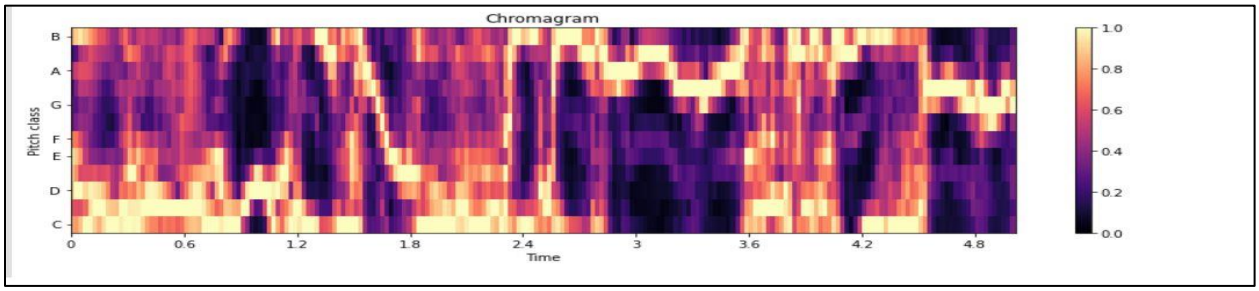
```
SAMPLE_RATE = 22050
fname_f =  '/content/drive/My Drive/Recordings/recordings/french2.wav'
y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)
C = librosa.feature.chroma_cqt(y=y, sr=sr)

# Make a new figure
plt.figure(figsize=(12,4))
# To make sure that the colors span the full range of chroma values, set vmin and vmax
librosa.display.specshow(C, sr=sr, x_axis='time', y_axis='chroma', vmin=0, vmax=1)
plt.title('Chromagram')
plt.colorbar()
plt.tight_layout()
```




```
SAMPLE_RATE = 22050
fname_f = '/content/drive/My Drive/Recordings/recordings/french2.wav'
y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)
C = librosa.feature.chroma_cqt(y=y, sr=sr)

# Make a new figure
plt.figure(figsize=(12,4))
librosa.display.specshow(C, sr=sr, x_axis='time', y_axis='chroma', vmin=0, vmax=1)
plt.title('Chromagram')
plt.colorbar()
plt.tight_layout()
```



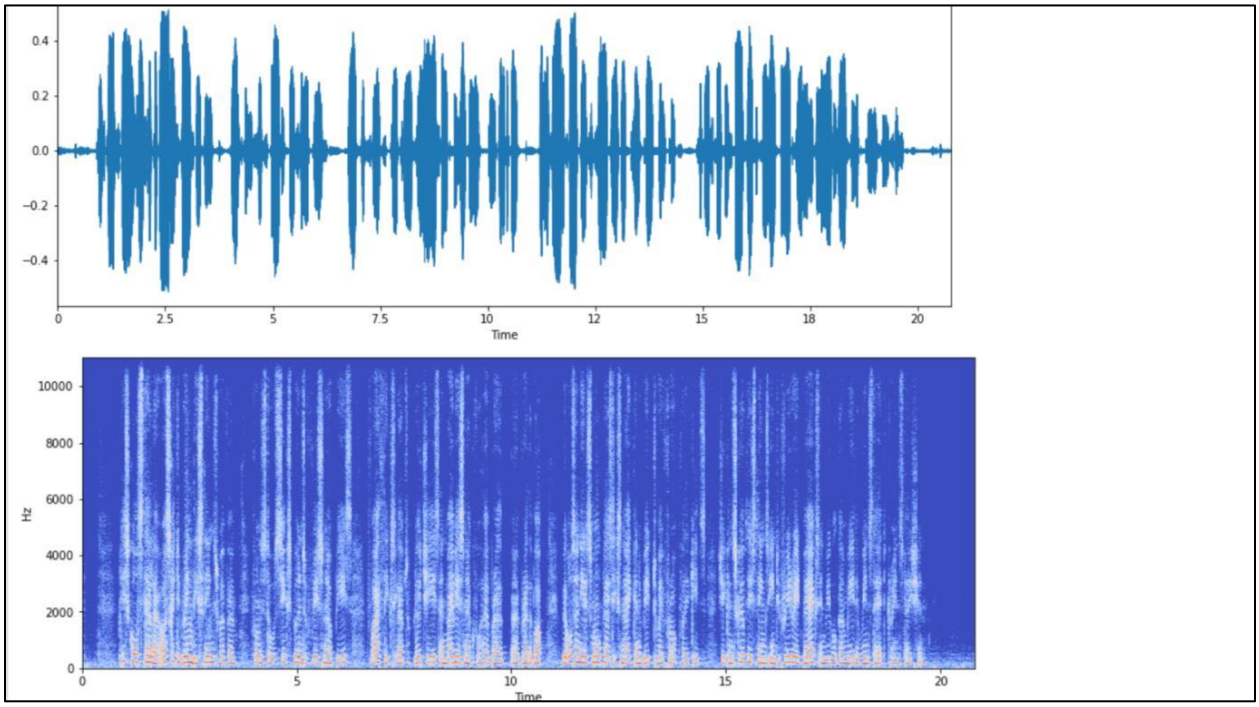
CNN(convolutional neural network)

When particular accent audio file is given as input it read the files and print the accuracy.

```
# Check the data of recordings data set
files=[]
files =  os.listdir('/content/drive/My Drive/Recordings/recordings')
print(files)
print (len([name for name in os.listdir('/content/drive/My Drive/Recordings/recordings') if os.path.isfile(os.path.join('recordings', name))]))
print(df.groupby("filename")['age'].describe().sort_values(by=['count'],ascending=False).head(10))
for i in range(0,2138):
  fname1 =os.listdir('/content/drive/My Drive/Recordings/recordings')[i]
  print(fname1)


x, sr = librosa.load('/content/drive/My Drive/Recordings/recordings/afrikaans1.wav')
print(x.shape)
print(sr)
plt.figure(figsize=(14, 5))
librosa.display.waveplot(x, sr=sr)
plt.show()
X = librosa.stft(x)
Xdb = librosa.amplitude_to_db(abs(X))
plt.figure(figsize=(14, 5))
librosa.display.specshow(Xdb, sr=sr, x_axis='time', y_axis='hz')
plt.show()


ipd.Audio('/content/drive/My Drive/Recordings/recordings/afrikaans1.wav')
```



```
[2]
import warnings
warnings.filterwarnings('ignore')
a = os.listdir('/content/drive/My Drive/Recordings/recordings')
# Read Data
data = pd.read_csv('/speakers_all.csv')
data.head(5)
print(data.shape)
data =data[32:]
valid_data = data[['age', 'birthplace','filename' ,'native_language','sex', 'speakerid','country']]
gender = {'male': 1,'female': 2}
valid_data.sex = [gender[item] for item in data.sex]

print(valid_data.shape)

y, sr = librosa.load('/content/drive/My Drive/Recordings/recordings/afrikaans1.wav', duration=2.97)
ps = librosa.feature.melspectrogram(y=y, sr=sr)
print(ps.shape)
librosa.display.specshow(ps, y_axis='mel', x_axis='time')
plt.show()

valid_data['path'] = '/' + valid_data['filename'].astype('str')

D = [] # Dataset

for row in valid_data.itertuples():
      y, sr = librosa.load('/content/drive/My Drive/Recordings/recordings' + row.path+'.wav', duration=2.97)
      ps = librosa.feature.melspectrogram(y=y, sr=sr)
      if ps.shape != (128, 128): continue
      D.append( (ps, row.sex) )
      print("Number of samples: ", len(D))
dataset = D
```

```
dataset = D
random.shuffle(dataset)



train = dataset[:2000]
test =  dataset[2000:]


X_train, y_train = zip(*train)
X_test, y_test = zip(*test)
# Reshape for CNN input
X_train = np.array([x.reshape( (128, 128, 1) ) for x in X_train])
X_test = np.array([x.reshape( (128, 128, 1) ) for x in X_test])

# One-Hot encoding for classes
y_train = np.array(keras.utils.to_categorical(y_train))
y_test = np.array(keras.utils.to_categorical(y_test))

num_classes = y_test.shape[1]
input_shape=(128, 128, 1)
model = Sequential()
model.add(Conv2D(24, (5, 5), strides=(1, 1), input_shape=input_shape,padding='same',activation='relu'))
model.add(Dropout(0.2))
model.add(Conv2D(48, (5, 5), activation='relu', padding='same'))
model.add(MaxPooling2D((4, 2), strides=(4, 2)))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

```
y_test = np.array(keras.utils.to_categorical(y_test))

num_classes = y_test.shape[1]
input_shape=(128, 128, 1)
model = Sequential()
model.add(Conv2D(24, (5, 5), strides=(1, 1), input_shape=input_shape,padding='same',activation='relu'))
model.add(Dropout(0.2))
model.add(Conv2D(48, (5, 5), activation='relu', padding='same'))
model.add(MaxPooling2D((4, 2), strides=(4, 2)))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(
  optimizer="Adam",
  loss="categorical_crossentropy",
  metrics=['accuracy'])

model.fit(
  x=X_train,
  y=y_train,
    epochs=3,
    batch_size=128,
    validation_data= (X_test, y_test))

score = model.evaluate(x=X_test,y=y_test)


print('Test accuracy:', score[1])
```

```
Number of samples:  2054
```

```
Number of samples:  2112
Number of samples:  2113
Number of samples:  2114
Number of samples:  2115
Number of samples:  2116
Number of samples:  2117
Number of samples:  2118
Number of samples:  2119
Number of samples:  2120
Number of samples:  2121
Number of samples:  2122
Number of samples:  2123
Number of samples:  2124
Number of samples:  2125
Number of samples:  2126
Number of samples:  2127
Number of samples:  2128
Number of samples:  2129
Number of samples:  2130
Number of samples:  2131
Number of samples:  2132
Number of samples:  2133
Number of samples:  2134
Number of samples:  2135
Number of samples:  2136
Number of samples:  2137
Number of samples:  2138
Train on 2000 samples, validate on 138 samples
Epoch 1/3
2000/2000 [==============================] - 211s 105ms/step - loss: 13.1930 - accuracy: 0.6085 - val_loss: 0.5219 - val_accuracy: 0.8116
Epoch 2/3
2000/2000 [==============================] - 208s 104ms/step - loss: 0.5909 - accuracy: 0.8115 - val_loss: 0.6514 - val_accuracy: 0.8768
Epoch 3/3
2000/2000 [==============================] - 209s 105ms/step - loss: 0.4580 - accuracy: 0.8650 - val_loss: 0.3440 - val_accuracy: 0.9203
138/138 [==============================] - 3s 24ms/step
Test accuracy: 0.9202898740768433
```

comparing all the accents.

1.In this project we are utilizing CNN way to deal with tackle issue set. (CNN) have been utilized for spectrographic examination of audio information for music grouping (Costa,Oliveira, and Silla 2017) and for recognition parts of human discourse.

2.The other accessible methodology is KNN, GMN anyway those drawn nearer doesn't

give satisfactory outcomes. K-Nearest Neighbor (KNN) is a regulated learning strategy where another occurrence is grouped dependent on the nearest preparing tests present in the component space. The proposed KNN model groups the given information signal is either discourse or music. Watchwords: Speech, Music, Feature Extraction, KNN.

KNN(k-nearest neighbours)

The sample code that we have worked on knn.

```
import keras
from keras.layers import Activation, Dense, Dropout, Conv2D, \
                         Flatten, MaxPooling2D
from keras.models import Sequential
import librosa
import matplotlib.pyplot as plt
import librosa.display
import numpy as np
import pandas as pd
import random
import os

import warnings

from sklearn.neighbors import KNeighborsClassifier
def print_results(predictions,data,desired_variable):
   print("Results:")
   print("Number of mislabeled points out of a total {} points : {}, performance {:05.2f}%"
   .format(
      test_data.shape[0],
      (data[desired_variable] != predictions).sum(),
      100*(1-(data[desired_variable] != predictions).sum()/data.shape[0])
      )
   )

np.random.seed(0)
sound_data = pd.read_csv("speakers_all.csv", encoding = "ISO-8859-1")
sound_data =sound_data[32:]
valid_data = sound_data[['filename' ,'sex']]
gender = {'male': 1,'female': 2}
factors_list = ["filename","sex"]
predictor_factors = ["sex"]
```

```
sound_data_norm = sound_data[factors_list]
data_sample = sound_data_norm.sample(frac=0.7, replace=False)
training_data = data_sample
test_data = sound_data_norm[~sound_data_norm.isin(training_data)].dropna()
knn_model = KNeighborsClassifier(n_neighbors=2)
knn_model.fit(training_data[predictor_factors],training_data["is_top_100"])
test_pred = knn_model.predict(test_data[predictor_factors])
print_results(test_pred,test_data)
```

6.We used Google colab to run code and we have uploaded recordings file to google drive and accessed through google colab.

ASSIGNING ID'S TO SPECIFIC COUNTRY FOR ACCENT RECOGNITION BASED ON COUNTRY

We have also tried to give specific ID to each country so that when we extract audio file it should read the accent and displays the ID assigned to that specific country.This model have some errors.

```
import keras
from keras.layers import Activation, Dense, Dropout, Conv2D, \
                         Flatten, MaxPooling2D
from keras.models import Sequential
import librosa
import matplotlib.pyplot as plt
import librosa.display
import numpy as np
import pandas as pd
import random
import os

import warnings
warnings.filterwarnings('ignore')
a = os.listdir('/content/drive/My Drive/Recordings/recordings')
# Read Data
data = pd.read_csv('/content/drive/My Drive/Recordings/speakers_all.csv')
data.head(5)
print(data.shape)
data =data[32:]
print(data['country'].isnull().sum())
data=data.fillna("na")
valid_data = data[['age', 'birthplace','filename' ,'native_language','sex', 'speakerid','country']]
desh = {'senegal':  1,
'cameroon': 2,
'nigeria' : 3,
'haiti':  4,
'usa':  5,
'jamaica':  6,
'liberia':  7,
'nicaragua':  8,
'south africa': 9,
'india':  10,
'sri lanka':  11,
'tajikistan' : 169,
'thailand': 170,
'tibet':  171,
'eritrea':  172,
'turkey': 173,
'turkmenistan': 174,
'vietnam':  175,
'benin':  176,
'na':177
}
valid_data.country = [desh[item] for item in data.country]
print(valid_data)
y, sr = librosa.load('/content/drive/My Drive/Recordings/recordings/afrikaans1.wav', duration=2.97)
ps = librosa.feature.melspectrogram(y=y, sr=sr)
print(ps.shape)
librosa.display.specshow(ps, y_axis='mel', x_axis='time')
plt.show()

valid_data['path'] = '/' + valid_data['filename'].astype('str')

D = [] # Dataset

for row in valid_data.itertuples():
      y, sr = librosa.load('/content/drive/My Drive/Recordings/recordings' + row.path+'.wav', duration=2.97)
      ps = librosa.feature.melspectrogram(y=y, sr=sr)
      if ps.shape != (128, 128): continue
      D.append( (ps, row.country) )
      print("Number of samples: ", len(D))
dataset = D
random.shuffle(dataset)



train = dataset[:2000]
test =  dataset[2000:]
```

```
# One-Hot encoding for classes
y_train = np.array(keras.utils.to_categorical(y_train))
y_test = np.array(keras.utils.to_categorical(y_test))

num_classes = y_test.shape[1]
input_shape=(128, 128, 1)
model = Sequential()
model.add(Conv2D(24, (5, 5), strides=(1, 1), input_shape=input_shape,padding='same',activation='relu'))
model.add(Dropout(0.2))
model.add(Conv2D(48, (5, 5), activation='relu', padding='same'))
model.add(MaxPooling2D((4, 2), strides=(4, 2)))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
model.compile(
  optimizer="Adam",
  loss="categorical_crossentropy",
  metrics=['accuracy'])

model.fit(
  x=X_train,
  y=y_train,
    epochs=3,
    batch_size=128,
    validation_data= (X_test, y_test))

score = model.evaluate(x=X_test,y=y_test)


print('Test accuracy:', score[1])
```

```
SAMPLE_RATE = 22050
fname_f =  '/content/drive/My Drive/Recordings/recordings/french1.wav'
y, sr = librosa.load(fname_f, sr=SAMPLE_RATE, duration = 5)
C = librosa.feature.chroma_cqt(y=y, sr=sr)

# Make a new figure
plt.figure(figsize=(12,4))
# To make sure that the colors span the full range of chroma values, set vmin and vmax
librosa.display.specshow(C, sr=sr, x_axis='time', y_axis='chroma', vmin=0, vmax=1)
plt.title('Chromagram')
plt.colorbar()
plt.tight_layout()
```

FUTURE SCOPE:

We can improvise CNN model to recognize accent.

We can also try to change model architecture to get better results.