Speaker Recognition

Group Name :- Error_404

Member 1 - Abinash Acharya (11840050)

Member 2 - Ganta Hemanthu Sai Kiran (11840500)

Project Report

1.Abstract

In this paper, a speaker recognition algorithm based on spectrograms is proposed. The spectrograms have been generated using Discrete Fourier Transform. Feature vector extraction has been done by using the row vector of the spectrograms using the spela library.

We have used **spela**, a package used to compute speech features e.g spectrograms and mel spectrograms implemented using tf.keras, to take advantage of GPU during computations.

We have planned to use 2 datasets, but for now we have trained and tested our model on one of them containing 7500 samples (audio clips) of 5 speakers and obtained a classification accuracy of 95% and 87% for spectrograms and mel spectrograms respectively.

2. Introduction

Speaker recognition, also known as voiceprint recognition, is an important branch of speech signal processing. It is a biometric identification technology that automatically detects a given speaker by extracting parameters representing his or her speech characteristics via a computer.

The goal of automatic speaker recognition systems is to extract, characterize and recognize the information in the speech signal for conveying speaker identity.

Speaker recognition is divided into two areas:

1. Speaker identification :- Deciding if a speaker is a specific person or is among a group of persons.
2. Speaker verification:- deciding if a speaker is who he/she claims to be.

Speaker verification is a 1:1 match where one speaker's voice is matched to one template whereas speaker identification is a 1: N match where the voice is compared against N templates

3. Problem Definition

Pattern matching in speech signals using spectrograms and Using a Deep learning Model to Identify the speaker from a group of N speakers.

4.Objective

The Steps We Will Take are as follows :

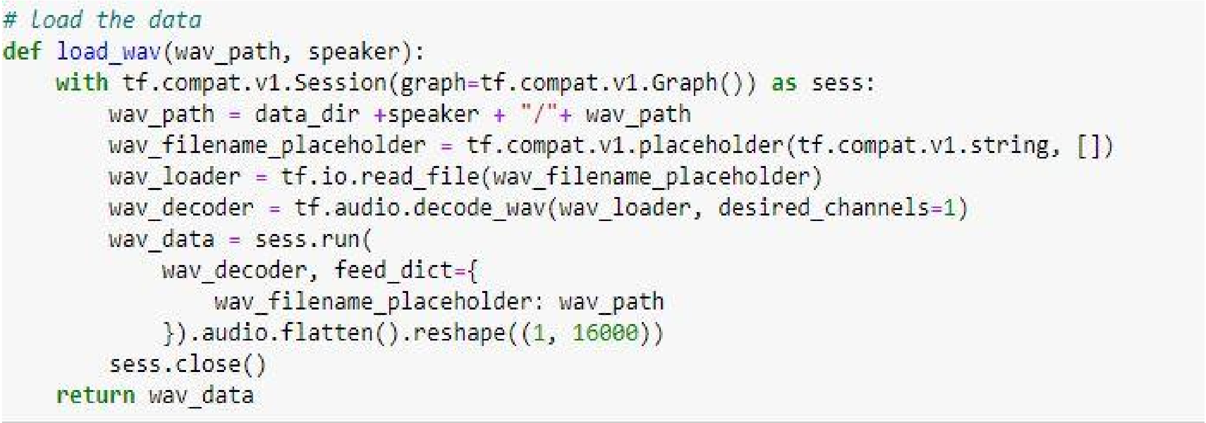
1. **Extract Audio Signals** :- The audio sampling rate used was 16000 Hz in the dataset on Kaggle. Thus, 16000 samples were extracted from each second of the audio clip.
2. **Spectrogram Generation** :- Discrete Fourier Transform was applied on the audio samples to create Spectrograms and Mel Spectrograms.

3. **Deep Learning Model** :- A Convolutional Neural Network (ConvNet/CNN) was applied with the spectrograms constituting the input-layer and speaker-labels as the output layer.

5.Technology Used

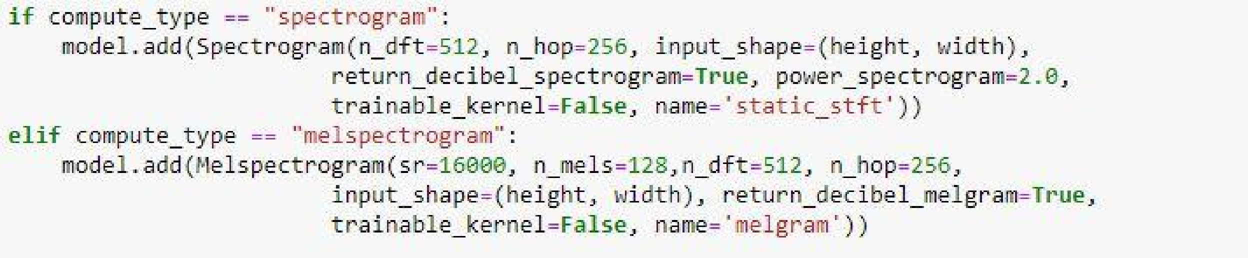
The various technologies used are :

1. **Tensorflow** :- We are using tensorflow for preprocessing audio files (.wav format) and converting them into signals. Apart from this , we are using Keras Sub-module in tensorflow for creating the convolution Neural Network.



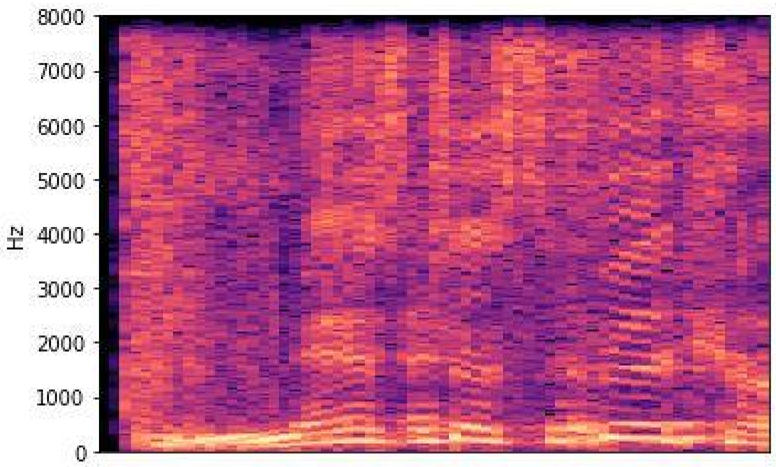
```python
# Load the data
def load_wav(wav_path, speaker):
    with tf.compat.v1.Session(graph=tf.compat.v1.Graph()) as sess:
        wav_path = data_dir +speaker + "/"+ wav_path
        wav_filename_placeholder = tf.compat.v1.placeholder(tf.compat.v1.string, [])
        wav_loader = tf.io.read_file(wav_filename_placeholder)
        wav_decoder = tf.audio.decode_wav(wav_loader, desired_channels=1)
        wav_data = sess.run(
            wav_decoder, feed_dict={
                wav_filename_placeholder: wav_path
            }).audio.flatten().reshape((1, 16000))
        sess.close()
    return wav_data
```

2. **Spela** :- We are using spela to convert audio signals into spectrograms and mel spectrograms.



```
if compute_type == "spectrogram":
    model.add(Spectrogram(n_dft=512, n_hop=256, input_shape=(height, width),
                          return_decibel_spectrogram=True, power_spectrogram=2.0,
                          trainable_kernel=False, name='static_stft'))
elif compute_type == "melspectrogram":
    model.add(Melspectrogram(sr=16000, n_mels=128,n_dft=512, n_hop=256,
                          input_shape=(height, width), return_decibel_melgram=True,
                          trainable_kernel=False, name='melgram'))
```

Below is depiction of the spectrogram plot from Nelson Mandela's Sound Clip (Created Using Librosa):



3. **Librosa:-** We are using librosa to plot the spectrograms and mel spectrograms for some of the audio clips.

6. Problems Faced

The major problem that we faced is the size of the Librispeech dataset. So, we decided to train/test our model on an alternative Kaggle Speaker Recognition Dataset for now. And Later on, if time permits, we can use the larger Librispeech dataset.

7.Datasets Used

The dataset we have used is Kaggle Speaker Recognition Dataset.

This dataset contains speeches of these prominent leaders:-

1. **Benjamin Netanyahu**
2. **Jens Stoltenberg**
3. **Julia Gillard**
4. **Magaret_Tarcher**
5. **Nelson Mandela**

The speaker's name also represents the folder names. Each audio in the folder is one-second 16000 sample rate PCM encoded.

The folder contains 1500 audio samples (each 1 second long) for each of the above speakers.

A folder called background_noise contains audios that are not speeches but can be found inside around the speaker environment e.g audience laughing or clapping. It can be mixed with the speech while training

8.Models Used

We created 2 models - one based on spectrograms and another based on mel spectrograms. Both are similar except for the input layer.

We used a convolution neural network model for training. Its layers are described below :-

1. **Input Layer** :- This consists of a 257 x 63 (128 x 63 in case of mel spectrogram) shaped two dimensional matrix consisting of spectrograms obtained from one-dimensional 16000 Hz audio signals.

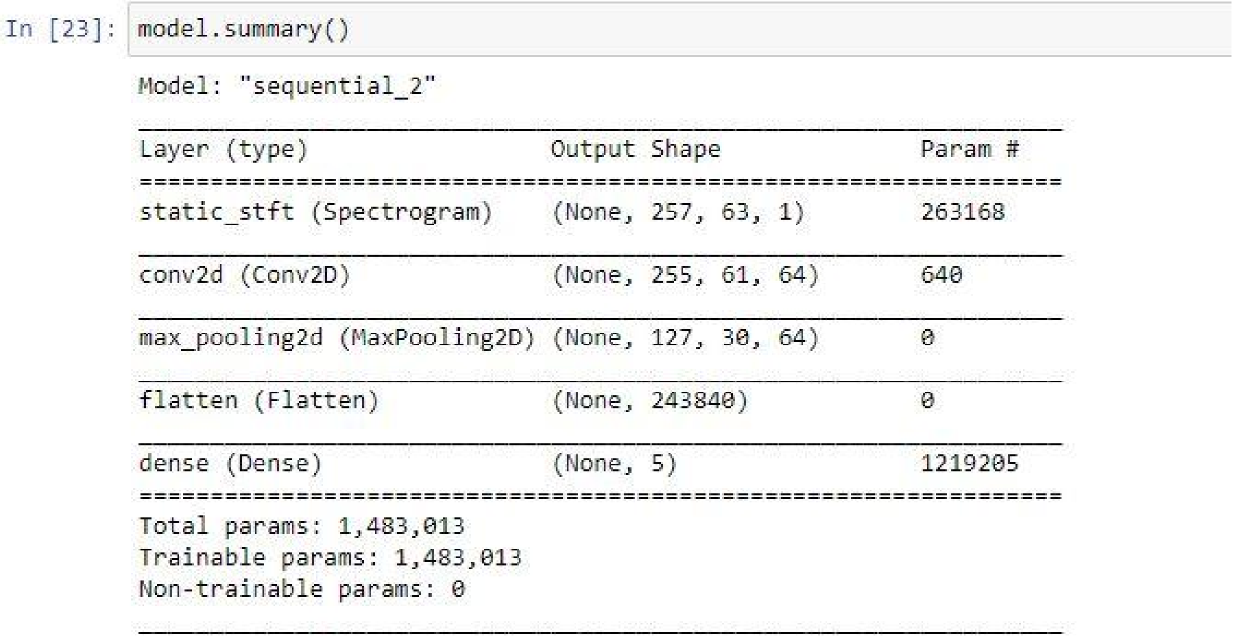
2. **Convolution Layer** :- This Layer consists of 64 filters having kernel size 3x3. and having a RELU activation function.

3. **Max-Pooling Layer :-** This Layer consists of max pooling matrix of size (2,2) and it is responsible for reducing the spatial size of the Convolved Feature and decreasing the computational power required to process the data through dimensionality reduction.

4. **Fully Connected Layer:-** This is used for learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. We obtain this layer by flattening out the previous layer's matrix.

5. **Output Layer:-** This layer consists of a 5 x 1 label (One hot encoding) for each training data point corresponding to each of the 5 speakers.

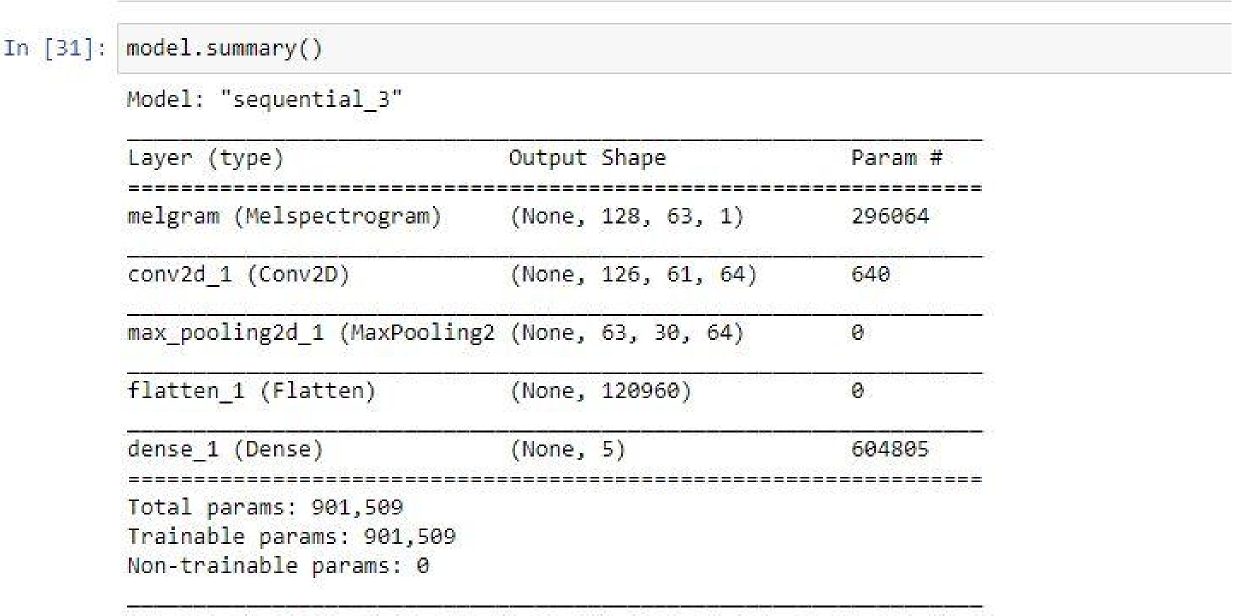
Spectrogram - Based Model



```
In [23]: model.summary()

         Model: "sequential_2"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         static_stft (Spectrogram)    (None, 257, 63, 1)        263168
         _________________________________________________________________
         conv2d (Conv2D)              (None, 255, 61, 64)       640
         _________________________________________________________________
         max_pooling2d (MaxPooling2D) (None, 127, 30, 64)       0
         _________________________________________________________________
         flatten (Flatten)            (None, 243840)            0
         _________________________________________________________________
         dense (Dense)                (None, 5)                 1219205
         =================================================================
         Total params: 1,483,013
         Trainable params: 1,483,013
         Non-trainable params: 0
         _________________________________________________________________
```

Mel Spectrogram Based Model



```
In [31]: model.summary()

         Model: "sequential_3"
         _________________________________________________________________
         Layer (type)                 Output Shape              Param #
         =================================================================
         melgram (Melspectrogram)     (None, 128, 63, 1)        296064
         _________________________________________________________________
         conv2d_1 (Conv2D)            (None, 126, 61, 64)       640
         _________________________________________________________________
         max_pooling2d_1 (MaxPooling2 (None, 63, 30, 64)        0
         _________________________________________________________________
         flatten_1 (Flatten)          (None, 120960)            0
         _________________________________________________________________
         dense_1 (Dense)              (None, 5)                 604805
         =================================================================
         Total params: 901,509
         Trainable params: 901,509
         Non-trainable params: 0
         _________________________________________________________________
```

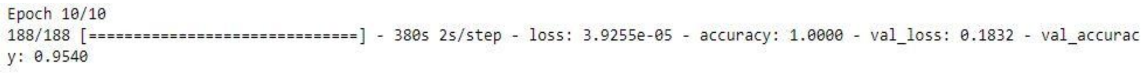
9.Implementation

We trained the model locally on the Kaggle Speaker Recognition Dataset mentioned above. Using sklearn, We split the dataset into 80-20 ratio and we used 80% of the audio clips for training and 20% of them for testing.

10.Result And Performance

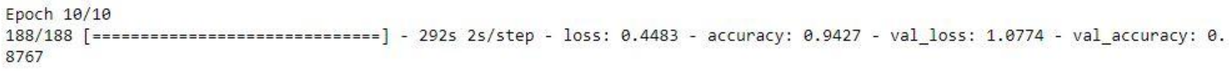
We have used 2 models :-

(i) Spectrogram-Based - This model uses spectrograms as the input layer. It gave an accuracy of 99% on the training dataset and 95% on the test dataset after 10 epochs.



```
Epoch 10/10
188/188 [==============================] - 380s 2s/step - loss: 3.9255e-05 - accuracy: 1.0000 - val_loss: 0.1832 - val_accurac
y: 0.9540
```

(ii) Mel Spectrogram-Based - This model uses mel spectrograms as the input layer. It gave an accuracy of 94% on the training dataset and 87% on the test dataset after 10 epochs.



```
Epoch 10/10
188/188 [==============================] - 292s 2s/step - loss: 0.4483 - accuracy: 0.9427 - val_loss: 1.0774 - val_accuracy: 0.
8767
```

11.Conclusion And Further Work

1. The model performed well on the Kaggle Speaker Recognition Dataset. So, we will try making a model for larger datasets like Librispeech consisting of audio clips of a larger number of speakers.

2. We are also planning to test our model on more noisy audio clips such that it can classify with reasonable accuracy irrespective of the noise.

3. We might also explore classifying speakers even when the audio clips are in different Languages other than English.

12.References

1. Speaker Recognition Based on Characteristic Spectrograms
2. Speaker Identification using Spectrograms of Varying Frame Sizes
3. Understanding Audio Features and spectrograms