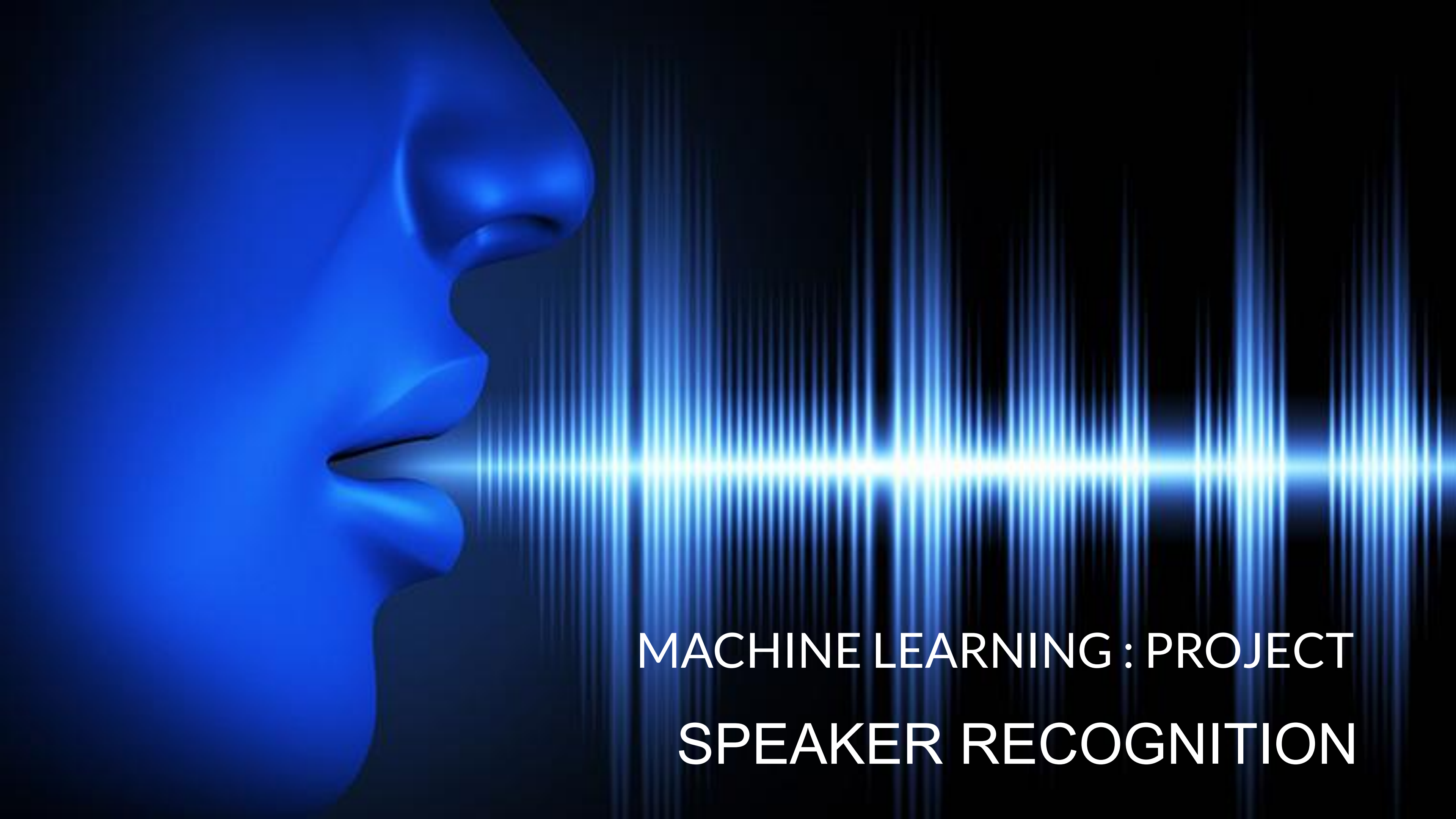


MACHINE LEARNING : PROJECT

SPEAKER RECOGNITION

ERROR_404

Members

Abinash Acharya (11840050)

Ganta Hemanth Sai Kiran (11840500)

Introduction

- Speaker Recognition is the intersection of 2 areas of computer science -
    1. Natural Language Technologies
    2. Biometrics
- Speaker recognition approach can be categorized into :

1. Speaker identification :- Determine which speaker (out of a group of speakers) a given voice sample belongs to.
2. Speaker verification :- Check if a voice sample belongs to a particular individual

Introduction - Motivation

1. Speech is one of the natural forms of communication

2. A person’s voice contain various parameters (like pitch etc.) that convey information such as emotion, gender, attitude, health and identity.

Here, we will try to identify a person based on his speech characteristics

Introduction - Applications

Speaker Recognition has various wise applications in multiple domains. Some of these are :

- Authentication
- Surveillance
- Forensic Speaker Recognition

Objective of our project

The objective of the project is to classify audio clips with respect to the speaker by training our model on a set of audio clips having the same set of speakers. The steps are as follows :

(i) Collect the voice samples

(ii) Pre-processing into audio signal vectors

(iii) Spectrogram and Mel Spectrogram extraction

(iv) Training Convolutional Neural networks on Spectrograms and mel spectrograms

(v) Predicting test samples using our model

Dataset Used

We have used a Speaker Recognition Dataset from Kaggle

It consists of audio samples from 5 speakers :

1. Nelson Mandela
2. Benjamin Netanyahu
3. julia Gillard
4. Margaret Thatcher
5. Jens Stoltenberg

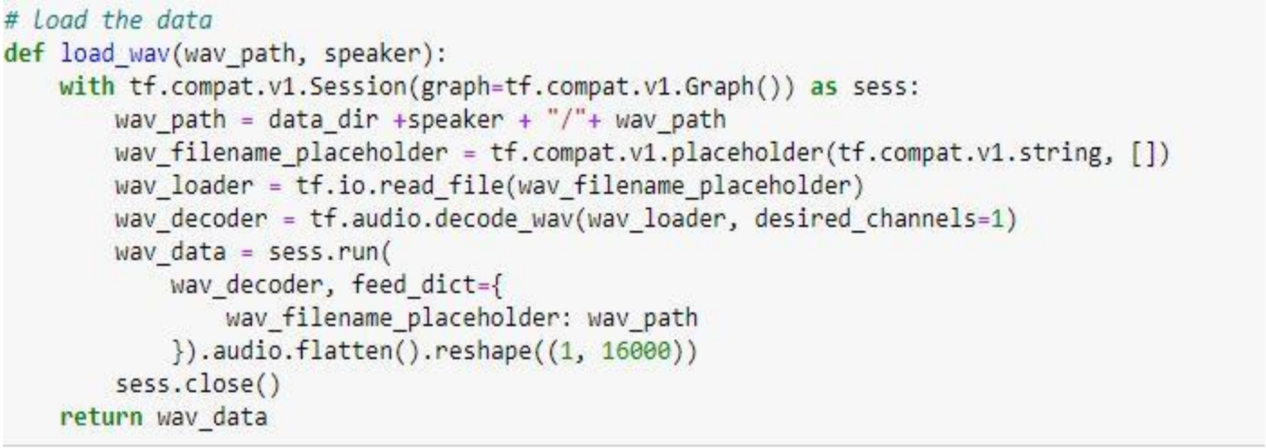
Audio sampled at - 16000 Hz

There are 1500 samples (1 second long) for each personality above.

Dataset - https://www.kaggle.com/kongaevans/speaker-recognition-dataset

Preprocessing Steps

1) **Extract Audio Signals** :- The audio sampling rate used was 16000 Hz in the dataset on Kaggle. Thus, 16000 audio signals were extracted from each second of the audio clip. This was done using the decode_wav method from Tensorflow.



```
# 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
```

Preprocessing Steps

2) **Spectrogram Generation** :- This can be done using band-pass filters (Analog signal processing) or by using Fourier transform (Digital Signal Processing)

Spectrogram :- A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time

Discrete Fourier Transform :- Digitally sampled data, in the time domain, is broken up into chunks, which usually overlap, and Fourier transformed to calculate the magnitude of the frequency spectrum for each chunk.

Here, we have used spela to convert audio signal vectors into spectrograms and mel spectrograms by using Discrete Fourier Transform.

Methodology

Models Available

1. Speaker Recognition With Sincnet
2. Transfer Learning
3. Spectrograms and Mel Spectrograms

Methodology

Brief Description of other models

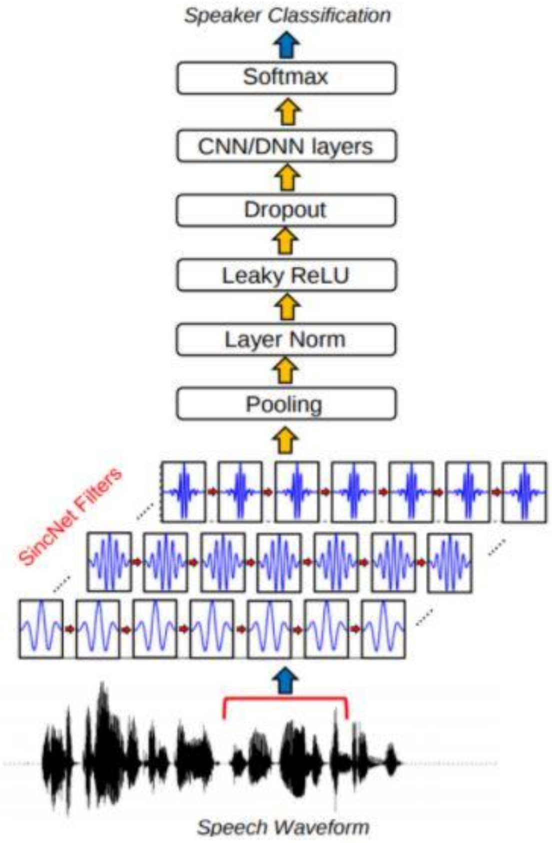
SincNet Model

1. Sinc(x)=sin(x)/x is used to implement Band-pass filters

2. $g[n,f_1,f_2] = 2f_2[\text{sinc}(2\pi f_2 n)] - 2f_1[\text{sinc}(2\pi f_1 n)]$

   where f_1=low cut-off frequency and f_2=high cutoff frequency

3. The low and high cutoff frequencies are the only parameters of the filter learned from data



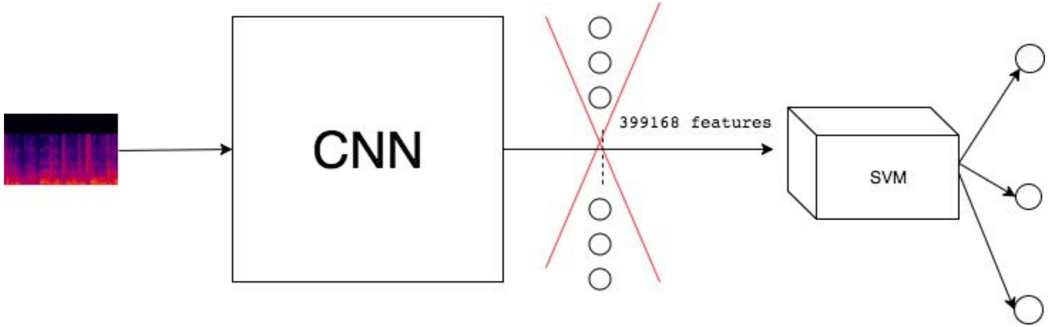
Methodology

Brief Description of other models

Transfer Learning

1. Convolution Layer as a feature extractor

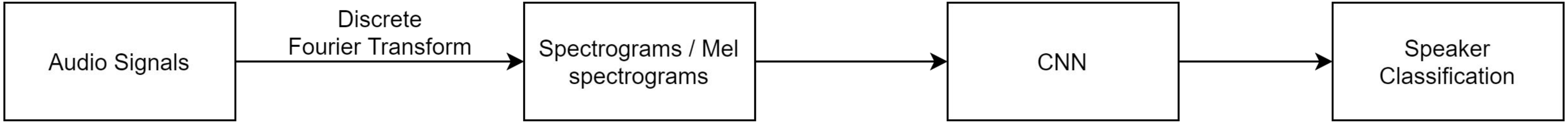
2. SVM as a classifier



Methodology

Model Chosen

Spectrograms with CNN



Audio Signals

Discrete Fourier Transform

Spectrograms / Mel spectrograms

CNN

Speaker Classification

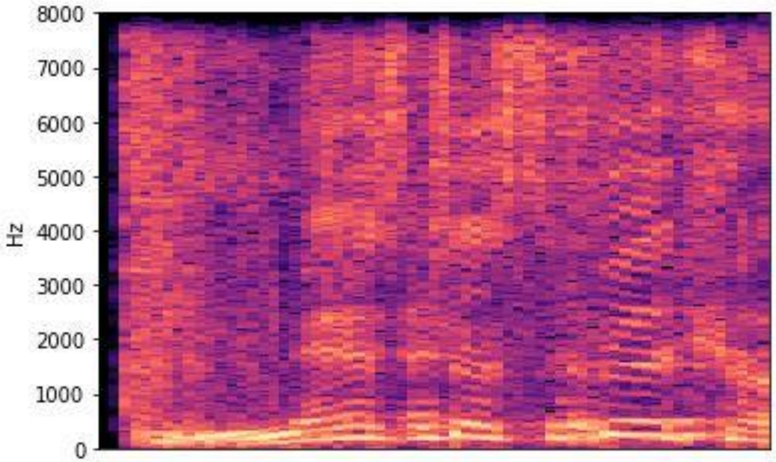
Methodology : Why This Model

- Traditionally, Mel-frequency cepstral coefficients (MFCCs) were used as speech features before the advent of Deep Learning.

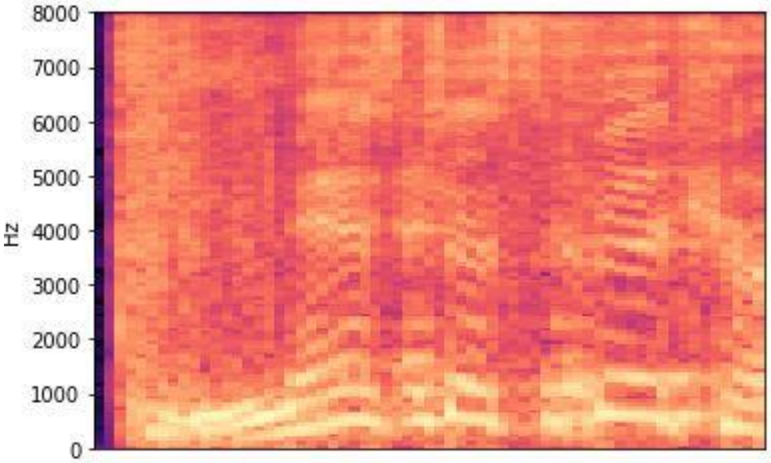
- But, Now we have CNN which can efficiently extract features from visual representations.

- Spectrogram is a visual representation of a sound wave. So, it can be take as input by the convolution neural network and the CNN network will extract features from the spectrograms on its own.

Methodology : Why This Model



Spectrogram

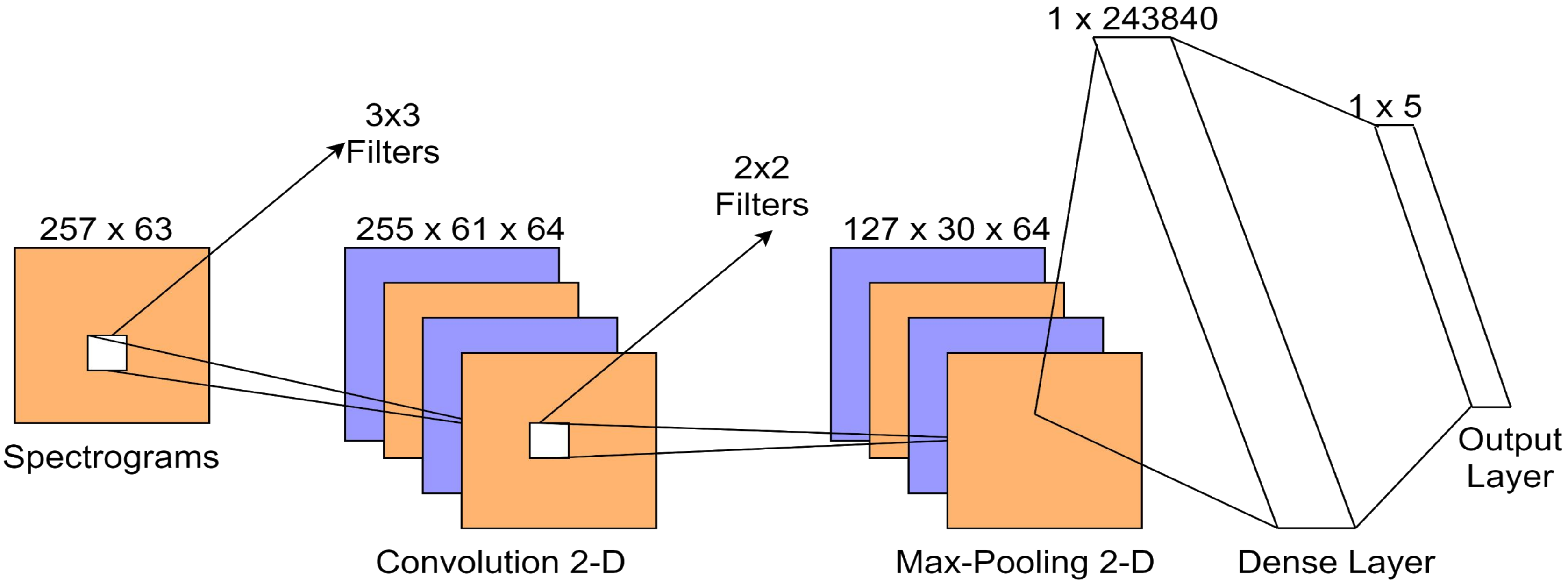


Mel Spectrogram

Methodology

Model Chosen

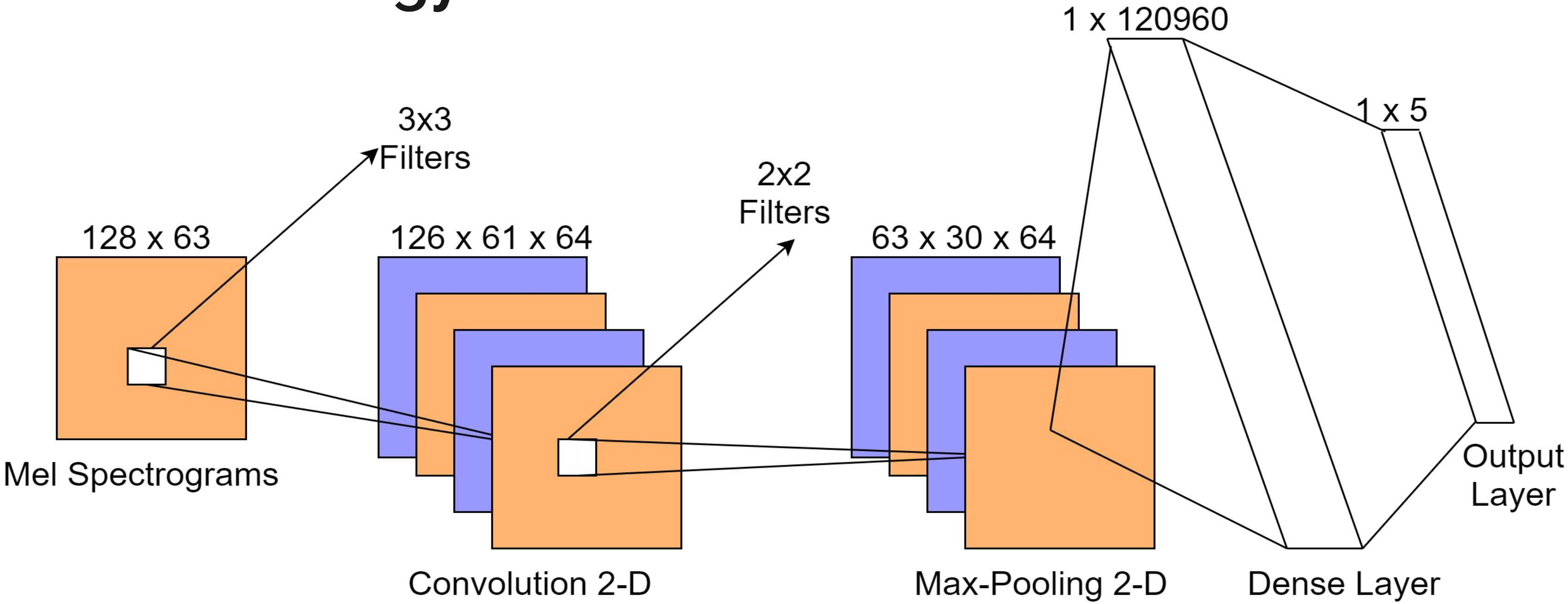
Architecture of the 1st Model



Methodology

Model Chosen

Architecture of the 2nd Model



Methodology : Why Both Spectrograms and mel spectrograms ?

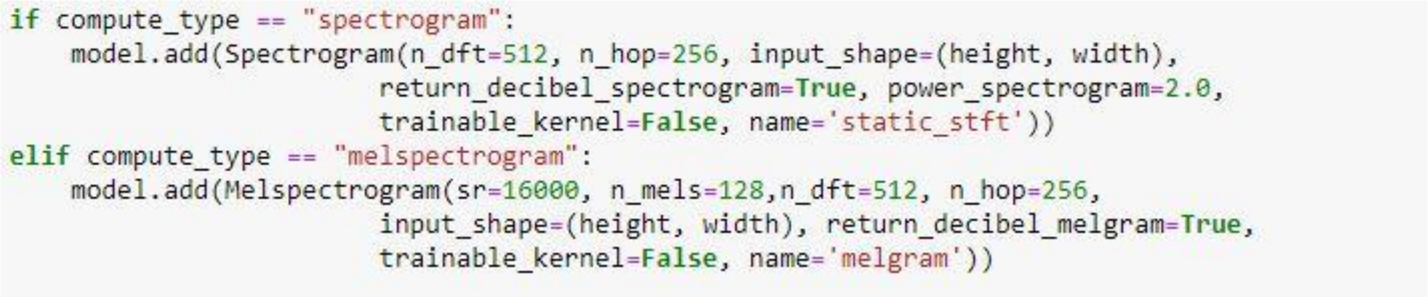
- Frequency in Spectrograms is represented in Hz Scale
- Hz is a linear scale in frequency
- But, Humans perceive frequencies logarithmically not linearly.
- That’s why we also use Mel Scale in our 2nd model which is a logarithmic scale.

Methodology

Model Chosen

Modules Used for the model

1. Spela :- For converting audio signals into spectrograms and mel spectrograms. This is the input layer of the model after preprocessing



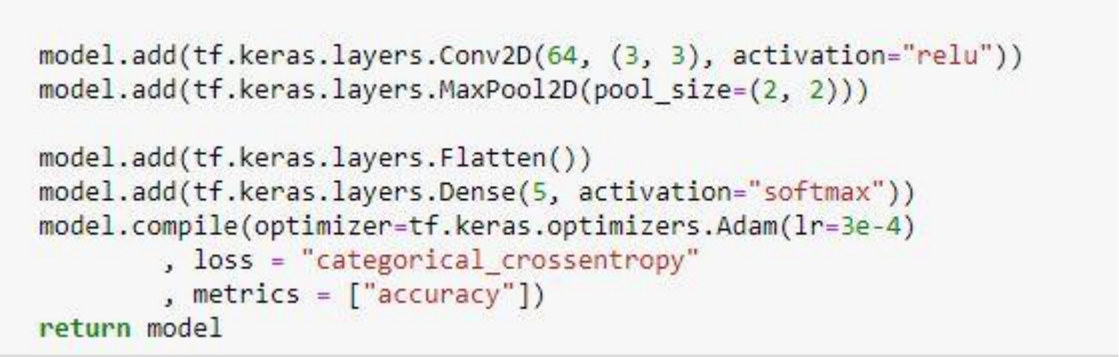
```
if compute_type == "spectrogram":
    model.add(Spectrogram(n_dft=512, n_hop=256, input_shape=(height, width),
                          return_decibel_spectrogram=True, power_spectrogram=2.0,
                          trainable_kernel=False, name='static_stft'))
elif compute_type == "melspectrogram":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

Methodology

Model Chosen

Modules Used for the model

2. Tensorflow :- For defining the layers of the Convolution Neural Network



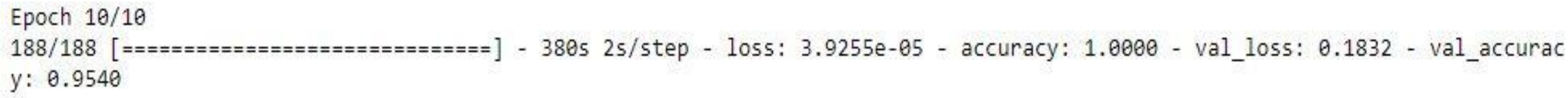
```
model.add(tf.keras.layers.Conv2D(64, (3, 3), activation="relu"))
model.add(tf.keras.layers.MaxPool2D(pool_size=(2, 2)))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(5, activation="softmax"))
model.compile(optimizer=tf.keras.optimizers.Adam(lr=3e-4)
        , loss = "categorical_crossentropy"
        , metrics = ["accuracy"])
return model
```

Results

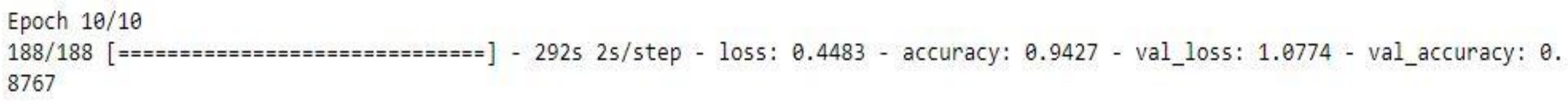
We have used 2 models and the results obtained are as follows :

1) 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
```

2) 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
```

Conclusion

- 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.

Future Direction

1) 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.
2) We might also explore classifying speakers even when the audio clips are in different Languages other than English.

References

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

Thank You !!!

Open For Questions

Team Error_404