STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

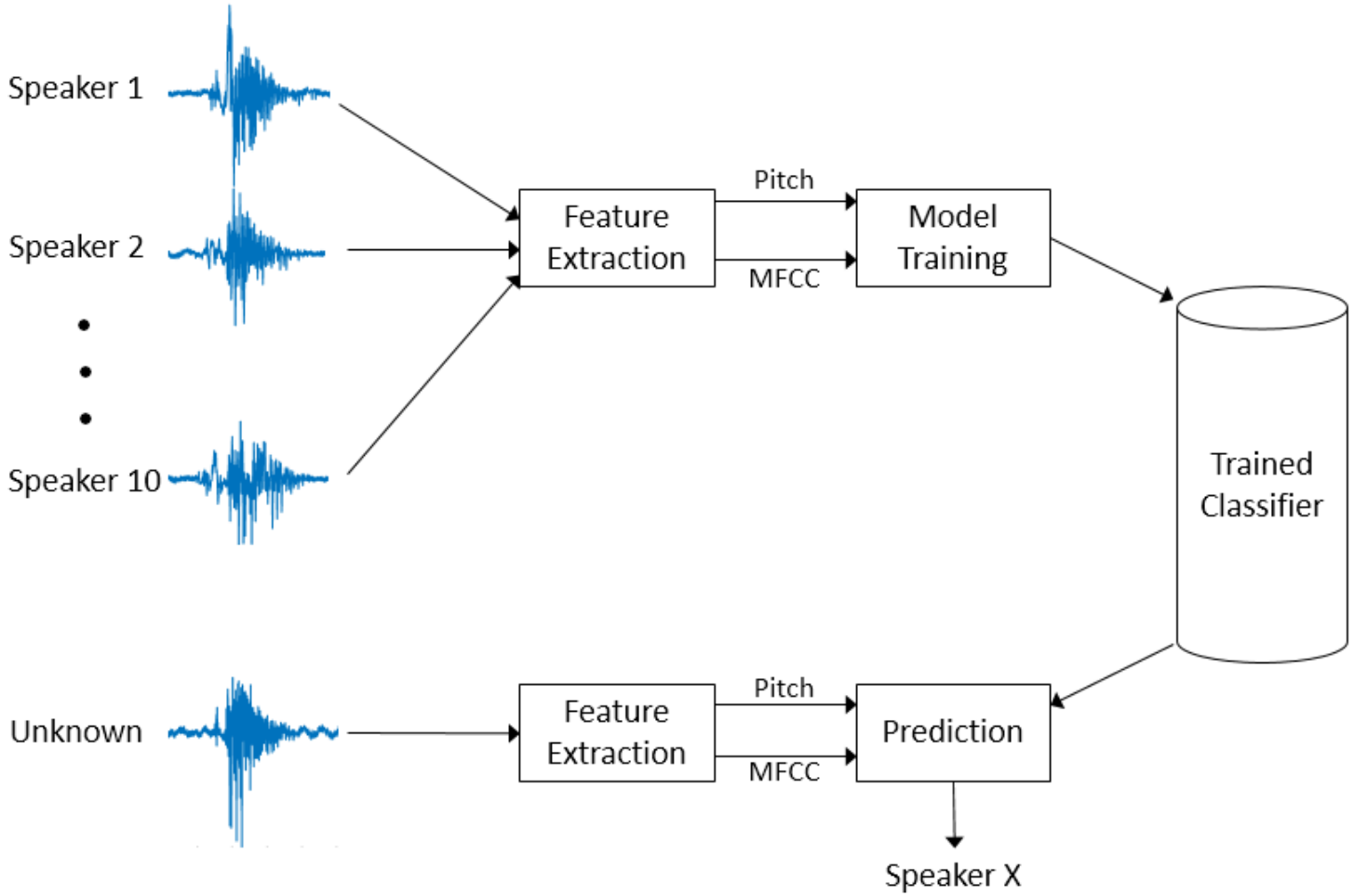
BIA 678 – Big Data Technologies

Daksh Shah
MEng in Applied Artificial Intelligence

Shrutik Pawale
MS in Data Science

Audio Language Detection

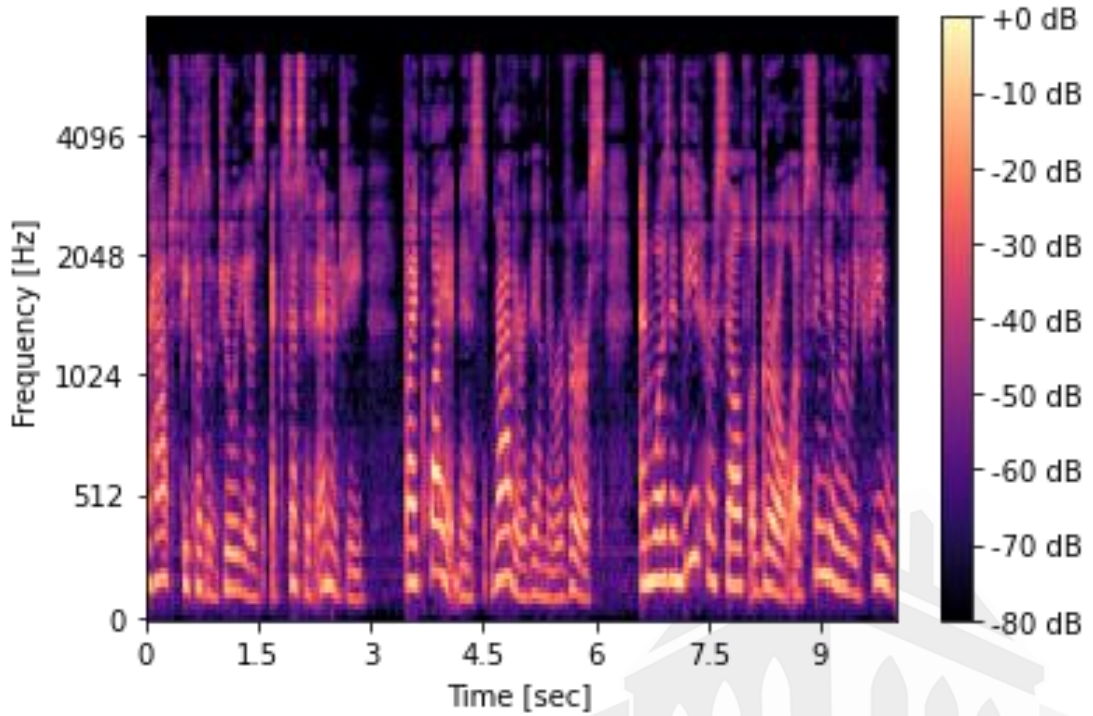
- The given dataset contains 10 second of speech recorded in English, German, and Spanish languages. Samples are equally balanced between languages, genders, and speakers.
- The core of the train set is based 73080 samples after applying several audio transformations (pitch, speed, and noise). No data augmentation has been applied. The number of unique speakers was increased by adjusting pitch (8 different levels) and speed (8 different levels).
- LibriVox recordings were used to prepare the dataset and particular attention was paid to a big variety of unique speakers since big variance forces the model to concentrate more on language properties than a specific voice.

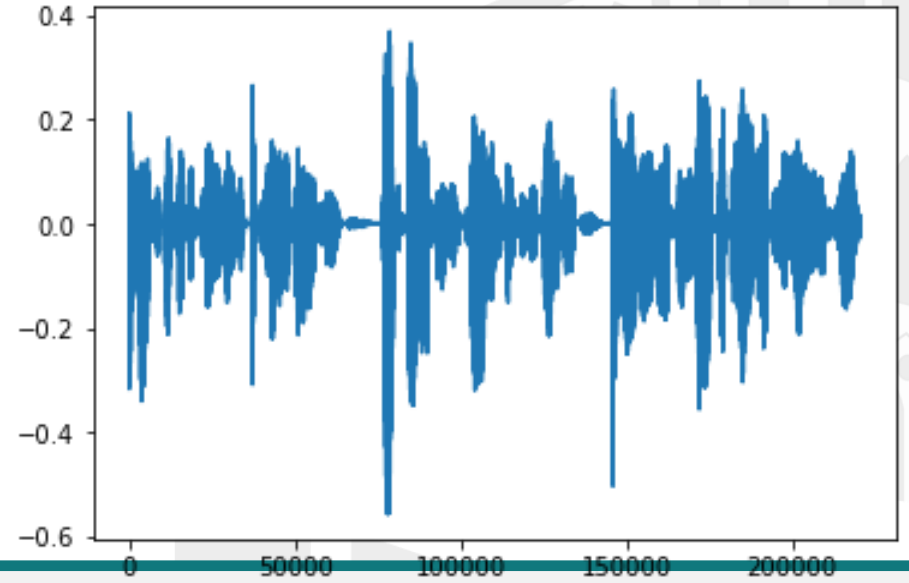


STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

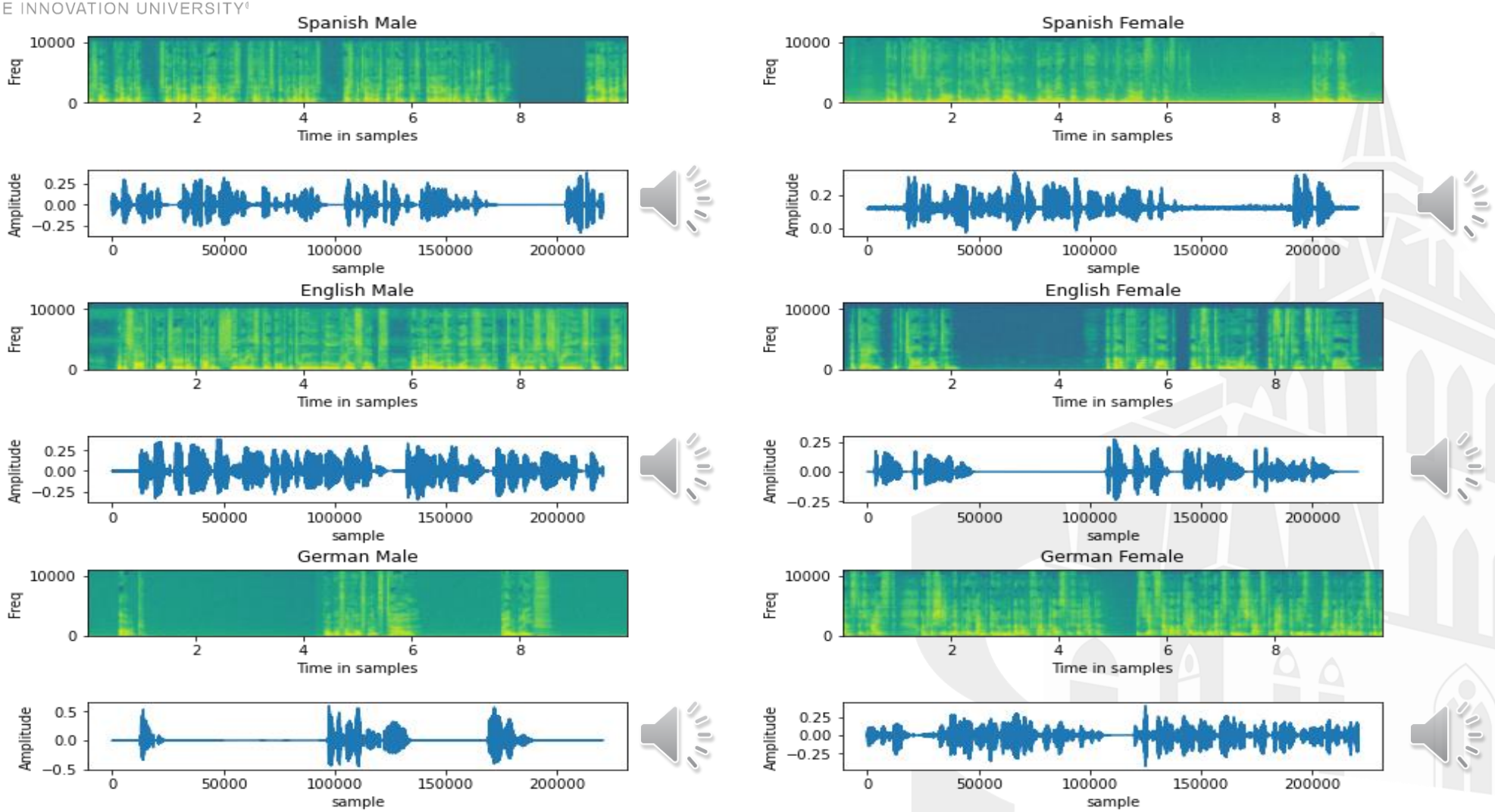
EDA

- The spectrogram implies how much the frequency of a signal changes over time.
- If the frequency matches at that second, then it will show less intensity.
- That is the reason, why there is a strip of dark line above 4096 Hz.
- This helps in identifying properties of nonlinear signals and that's why it is helpful in analyzing real-world data with a lot of frequency components and noise.





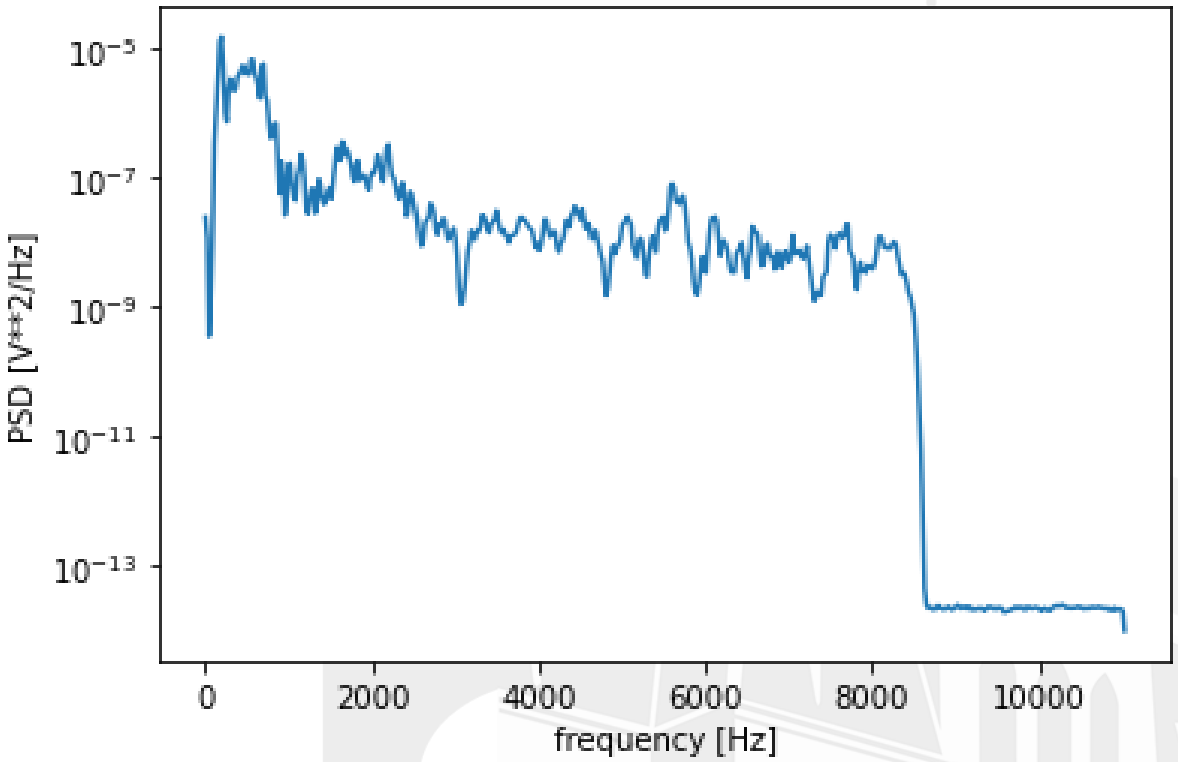
Sample Data


Spanish Male
Freq
Time in samples
Amplitude
sample
Spanish Female
English Male
English Female
German Male
German Female

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

EDA

- This tells us the power of a signal at different frequencies.
- In other words, it shows at which frequencies, variations are strong and at which frequencies, variations are weak.
- From the plot, higher frequencies contain less power as it is out of our normal vocal range.



STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Data Processing

We can extract the following features from the Audio File Name Format:
"(language)_(gender)_(recording_ID).fragment(index)[.(transformation)(index)].flac"

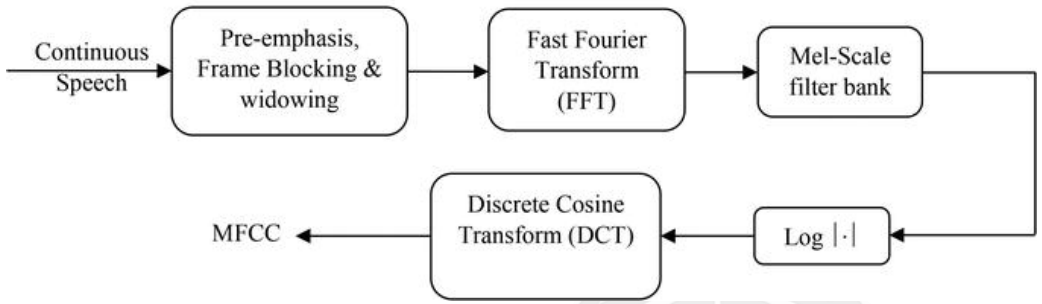
	filename	lang	gender	user_id	fragment	edit
	train\es_f_1d27c6d589eeff17973ffd0b7a77a70a.fr...	es	f	1d27c6d589eeff17973ffd0b7a77a70a	fragment5	speed5
	train\es_f_53b555eab2b3baada380f7d3ede20b20.fr...	es	f	53b555eab2b3baada380f7d3ede20b20	fragment14	pitch4
	train\de_f_d94712992f41e3d8d21f22274b3d8fd9.fr...	de	f	d94712992f41e3d8d21f22274b3d8fd9	fragment24	noise6
	train\en_f_10134f409d9b7b0b95fed6e025febcad.fr...	en	f	10134f409d9b7b0b95fed6e025febcad	fragment25	noise7
	train\es_m_b8e0e6f56f02e6f8f79cc360958e5982.fr...	es	m	b8e0e6f56f02e6f8f79cc360958e5982	fragment8	noise4
	...	...	...	...	...	...
	train\es_m_d5b91a4ffb1ead826b7968ec19cbfa1c.fr...	es	m	d5b91a4ffb1ead826b7968ec19cbfa1c	fragment3	noise10
	train\de_m_fc6bd6bb9d66a89bb8d8a8a7efa23e6b.fr...	de	m	fc6bd6bb9d66a89bb8d8a8a7efa23e6b	fragment4	noise6
	train\de_m_d22535879801cc9c4452d9ed9de5bf61.fr...	de	m	d22535879801cc9c4452d9ed9de5bf61	fragment20	speed4
	train\de_f_2825fa225d6ca4800f0cf0504b76ca65.fr...	de	f	2825fa225d6ca4800f0cf0504b76ca65	fragment11	pitch6
	train\es_f_bf4285930fa46f2052e5bdbc37a8a4df.fr...	es	f	bf4285930fa46f2052e5bdbc37a8a4df	fragment21	speed2

After reading the audio file using the sound file python package, we get the sample rate and an array of 220500 numbers. The resulting array contains the audio data as a sequence of samples, where each sample represents the amplitude of the audio signal at a specific point in time.

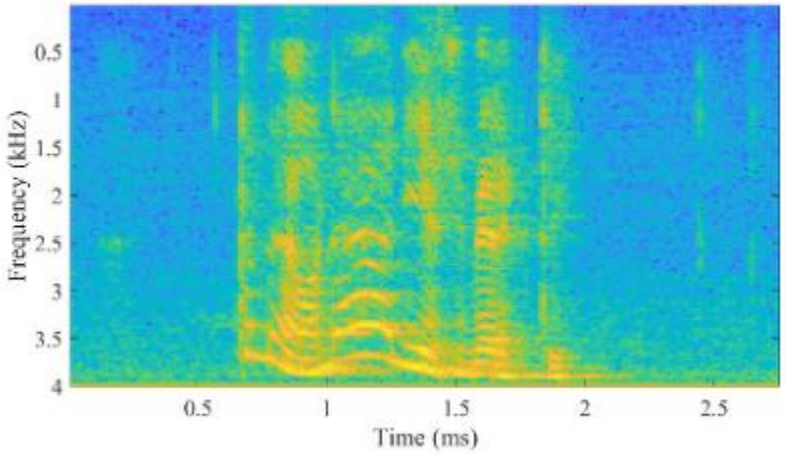
	0	1	2	3	4	5	6	7	8	9	...	220497	220498	220499	sample_rate
0	-0.020905	-0.031860	-0.028931	-0.020203	-0.002075	0.010193	0.013611	0.004120	-0.009247	-0.012665	...	0.000000	0.000000	0.000000	22050
1	-0.007965	-0.007202	0.003601	0.010895	0.014069	0.005890	-0.005829	-0.019653	-0.028564	-0.035706	...	0.018311	0.018646	0.011688	22050
2	-0.070190	-0.066132	-0.063599	-0.062103	-0.059601	-0.055115	-0.050598	-0.046295	-0.043060	-0.038849	...	0.015778	0.015717	0.017517	22050
3	-0.005951	-0.011993	-0.009888	-0.012848	-0.014374	-0.015961	-0.013062	-0.013824	-0.015961	-0.020050	...	-0.001495	-0.006683	-0.006561	22050
4	0.001556	0.001404	0.001617	0.002106	0.002625	0.002869	0.001709	0.000946	0.001740	0.002380	...	-0.038422	-0.038666	-0.038940	22050
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
495	-0.016266	-0.013611	0.000397	0.020569	0.032318	0.035248	0.035309	0.038361	0.045197	0.046844	...	-0.001801	0.019928	0.016479	22050
496	0.000122	0.001282	0.002045	0.002472	0.002960	0.003479	0.003418	0.002960	0.002747	0.002960	...	0.015900	0.013397	0.016937	22050
497	0.012054	0.015717	0.015717	0.018982	0.018433	0.018677	0.016052	0.018707	0.022522	0.032166	...	-0.009430	-0.018799	-0.018768	22050
498	-0.002380	0.003052	0.003204	-0.004486	-0.004395	0.001587	0.006195	-0.012115	-0.008118	-0.005341	...	0.000000	0.000000	0.000000	22050
499	0.006256	-0.006653	-0.027435	-0.017883	0.003571	0.021820	0.022003	0.009186	-0.001740	-0.008087	...	-0.007996	-0.004730	-0.002106	22050

Feature Extraction - MFCC

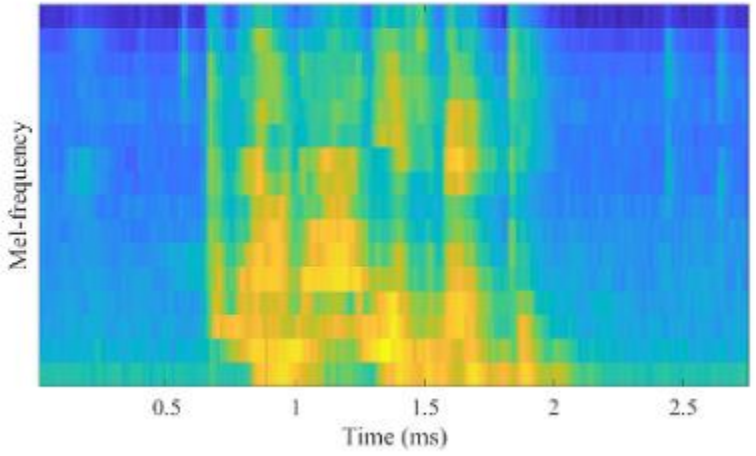
MFCC (Mel-Frequency Cepstral Coefficients) is a technique used in signal processing to analyze and represent the sound of a human voice or other sound signals. The sound is first broken down into many tiny pieces called frames, and then for each frame, the MFCC algorithm measures the power of different frequency bands within that frame. The frequency bands are spaced out in a way that is more like how the human ear perceives sound, which is why it's called Mel-frequency. Next, the algorithm applies some math operations to these frequency band measurements to reduce the dimensionality and capture the most important features of the sound. The resulting features are called cepstral coefficients and they can be used as inputs to machine learning models for tasks like speech recognition or music genre classification.



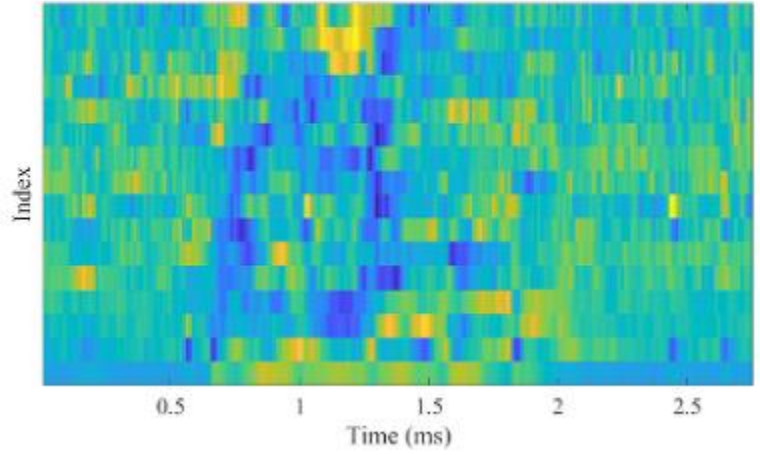
Spectrogram of a segment of speech



Spectrogram after multiplication with mel-weighted filterbank

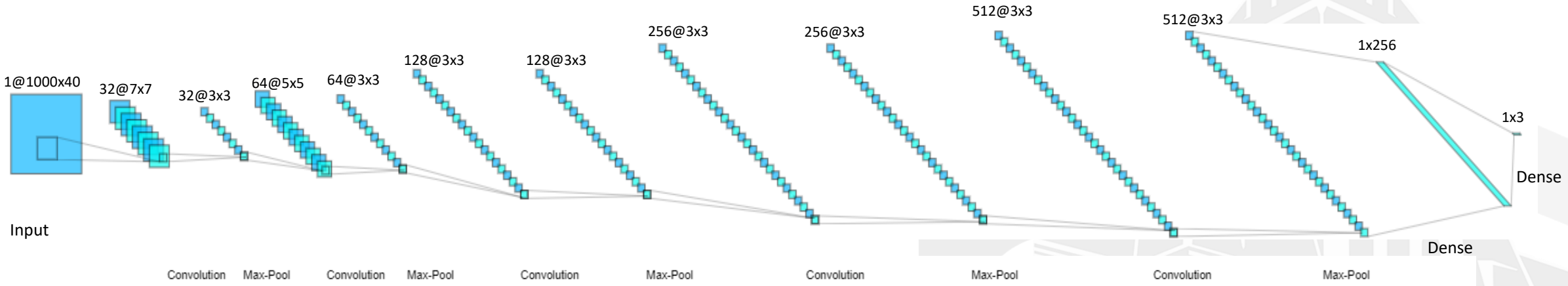


Corresponding MFCCs



Model Training - Convolution Neural Network

MFCC Output Dimensions:
12800 x (1000x40)



***Batch Normalization between every steps*

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Model Results

Train Set

```
In [72]: 1 confusion_matrix(y_train1,y_pred1)

Out[72]: array([[4262,    4,    8],
                [   0, 4256,    0],
                [   3,    0, 4267]], dtype=int64)
```

```
In [73]: 1 print(classification_report(y_train1,y_pred1))
```

	precision	recall	f1-score	support
de	1.00	1.00	1.00	4274
en	1.00	1.00	1.00	4256
es	1.00	1.00	1.00	4270
accuracy			1.00	12800
macro avg	1.00	1.00	1.00	12800
weighted avg	1.00	1.00	1.00	12800

Test Set

```
In [75]: 1 confusion_matrix(y_test1,y_pred2)

Out[75]: array([[1054,    5,    9],
                [   5, 1055,    4],
                [   2,    1, 1065]], dtype=int64)
```

```
In [76]: 1 print(classification_report(y_test1,y_pred2))
```

	precision	recall	f1-score	support
de	0.99	0.99	0.99	1068
en	0.99	0.99	0.99	1064
es	0.99	1.00	0.99	1068
accuracy			0.99	3200
macro avg	0.99	0.99	0.99	3200
weighted avg	0.99	0.99	0.99	3200

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY

Thank You