Speech Emotion Recognition

Data Science Final Project

Bach, Kien, Sauryanshu and Sike

LAYOUT

- Dataset
- Exploring and Augmenting the Data
- Feature Extraction
- Baseline Models
- More Advanced Models
- Comparison of the Models
- Model Evaluation

PROBLEM STATEMENT

Can we predict human emotion in speech?

DATA

DATA

- 1440 individual audio files = 1440 observations
- Spread evenly among 24 Voice actors, with 60 trials per actor.
- Gender balanced and Lexically matched statements.
- Either of the two statements:
  - Dogs are sitting by the door
  - Kids are talking by the door

FEATURES

- All individual audio files had 7 features:
  - Modality: AV or Audio Only
  - Vocal Chanel: Speech/ Song
  - Emotion: Neutral, Calm, Happy, Sad, Angry, Fearful, Disgust, Surprised
  - Emotional intensity: Normal, Strong
  - Statement: Kids are..., Dogs are...
  - Repetition: 1^{st} Repetition, or 2^{nd} Repetition
  - Actor: Male or Female
- Emotion: Label
- All of these could be our labels. Why? Because we use a Neural Network
- Using a Neural network means that majority of our features are rendered irrelevant.

Samples

- Anger
- Fear
- Happy
- Sad

Data Augmentation

Objectives: Prevent overfitting, increase training set, increase test accuracy

Examples of Augmentation techniques

Stretching

Pitch

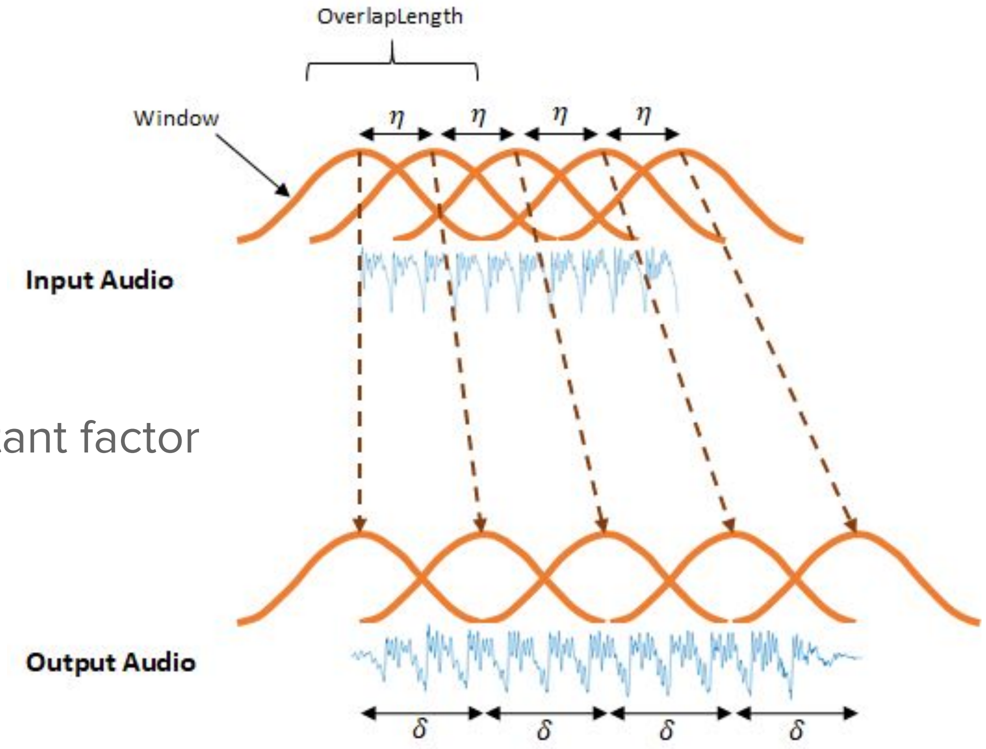
Shifting

Stretching

- Stretch time

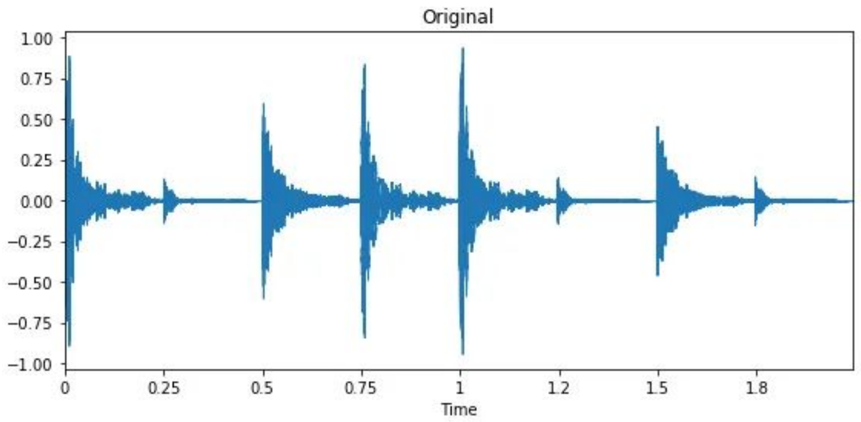
- Frequency modulated by a constant factor determined by time stretch

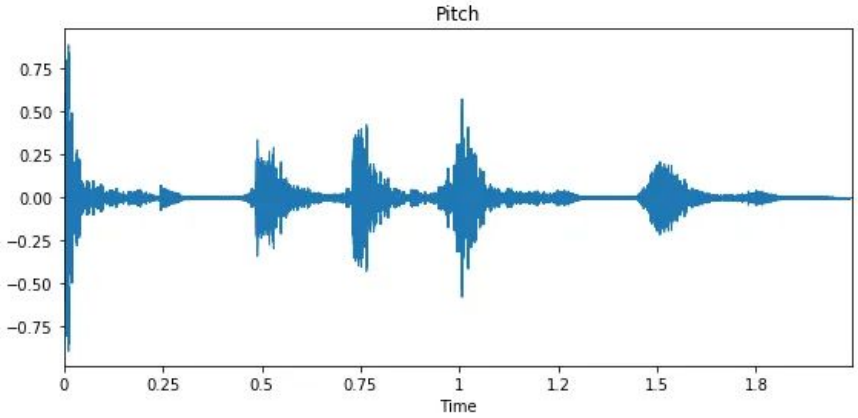
- Maintain proportionality between amplitudes



Pitch

- Two types of pitch Shifts:
  - Time and Frequency
- We use Frequency
- Change frequency randomly





Augmented Audio Samples

- Shifting
- Stretching
- Pitch
- Noise Injection

Feature Extraction

What sort of audio features can we extract from the audio file?

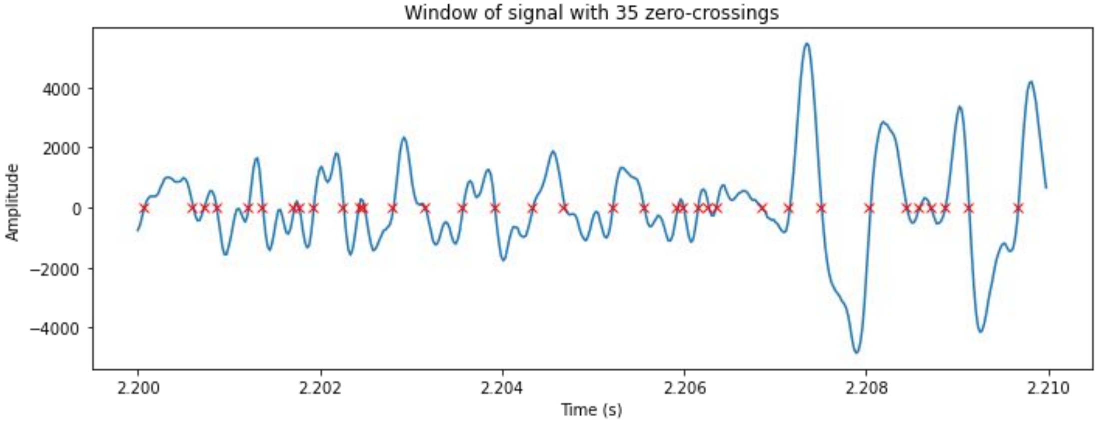
Feature Extraction

- Remember! We’re using a neural network. Since we don’t specify form of the model in a NN, we only provide features for it to train on.

- Several possibilities:
    - Mel Frequency Cepstral Coefficients (MFCC)
    - Zero Crossing Rate
    - Chroma Features

Zero Crossing Rate

- Notes the number of times The discrete audio values change signs (+ to - and vice versa)
- Not particularly useful for speech recognition



Chroma Features

- A broad range of specific features fall within Chroma Features, such as Chroma Vector, Chroma Stft

All of them focus on pitch-level changes in the audio Data

Mel Frequency Cepstral Coefficients

- Extraction of MFCCs are quite math-heavy, and complex

- But in essence, they extract all essential elements of an audio:
    - Frequency changes, amplitude changes, et cetera.

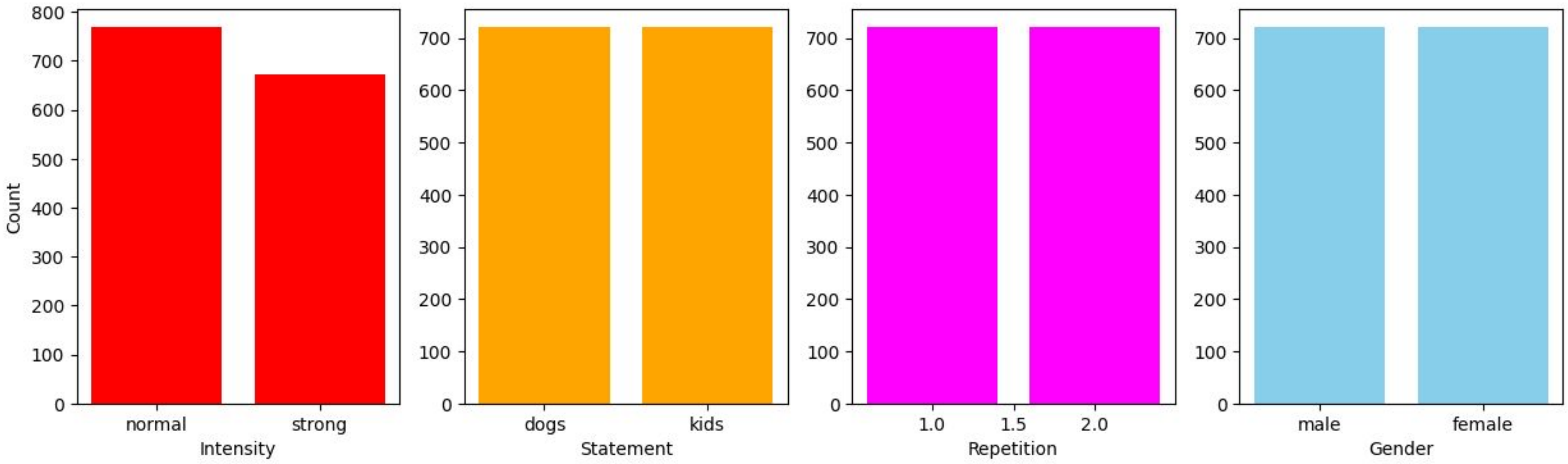
- Highly capable for Voice recognition Models

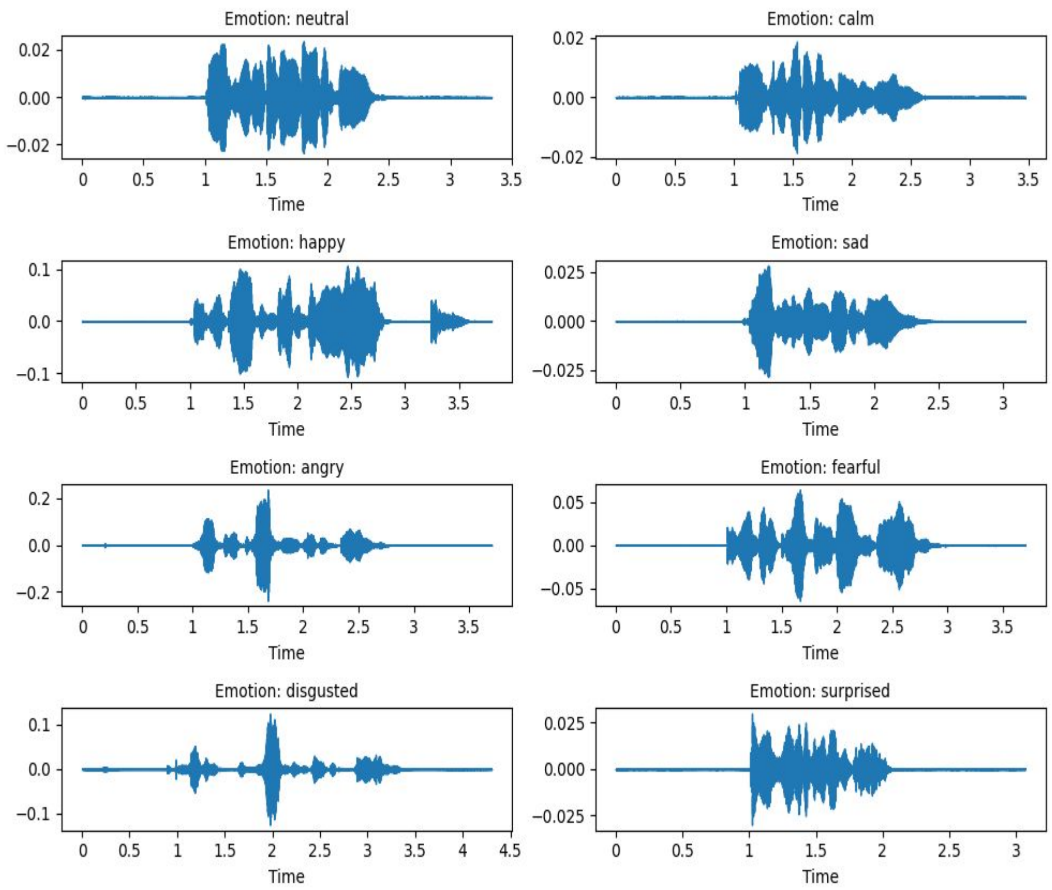
- We tried other features, but given MFCC's performance, we chose to use MFCCs as a feature.

Data Exploration

What do our audio files look like?

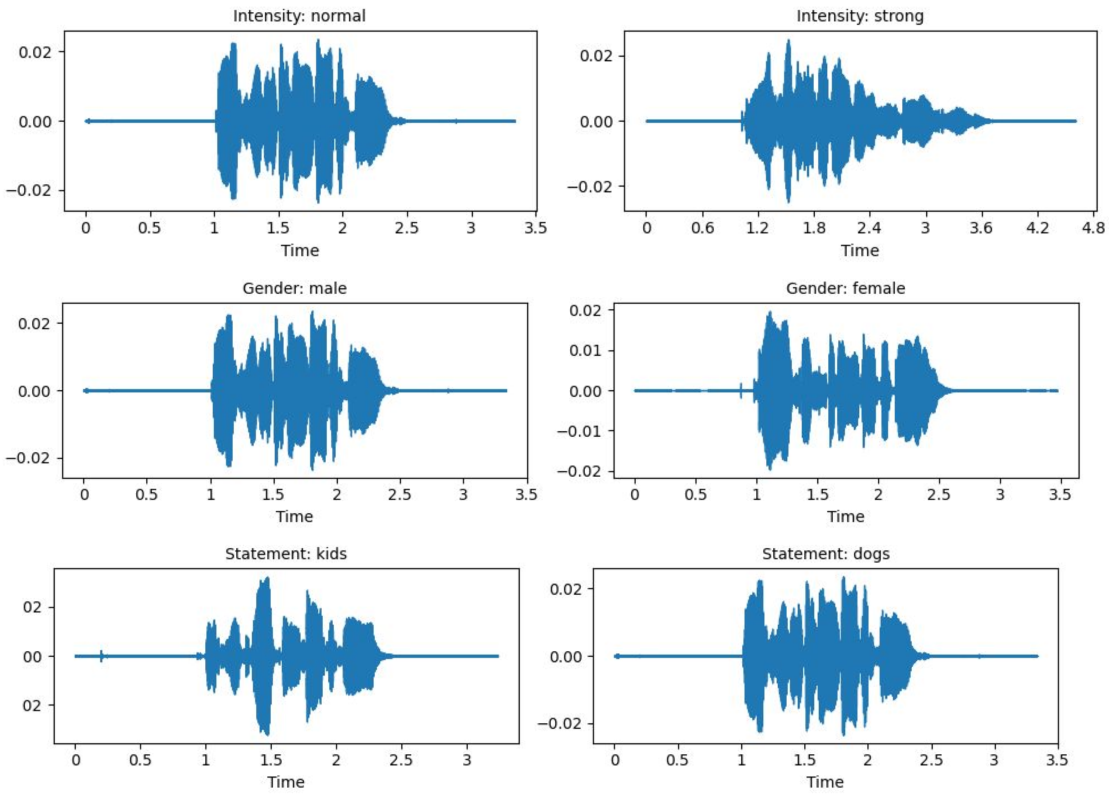
Generally well-balanced across attributes!





- Extracting MFCC from these audio files
- Spectrograms show enough variability amongst target labels on Amplitude, Frequency, duration, and changes within them.
- If no variability amongst them, NN would be useless

More Examples from across our non-label features



MODELS

Baseline Models

K Nearest Neighbours

Multilayer Perceptron

Convolutional Neural Network

K Nearest Neighbours

- Cross-validation was performed on different k values using ten splits
- k = 1 makes for the best number of neighbors, as F1 steeply drops as k increases
- Achieved an accuracy of 55%

Multilayer Perceptron

- Capable of learning non-linear relationships, which is critical in handling the complexities of human speech
- Includes hidden layers, allowing it to learn a hierarchy of features
- Our model had 14 hidden layers and 131,688 trainable parameters, and achieved an accuracy of 56%.

Convolutional Neural Network

- Highly effective at recognizing patterns in spatial data. In speech recognition, converting audio into spectrograms transforms the problem into a 2D image recognition task
- Can automatically learn necessary features of raw data
- Our model has 18 hidden layers and 110,216 trainable parameters, achieving an accuracy of 45%

Advanced Models

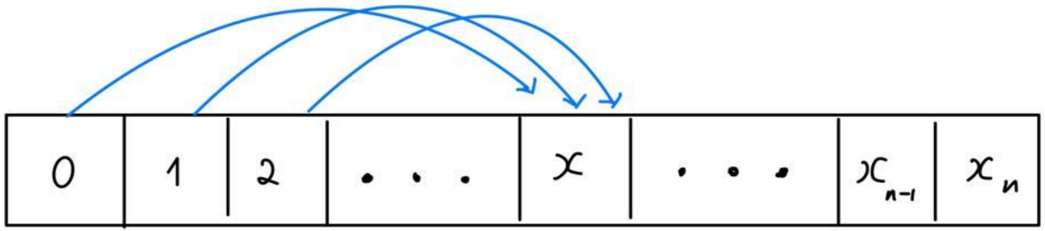
Long-Short Term Memory Network

CNN + LSTM Network

Bi-directional LSTM Network

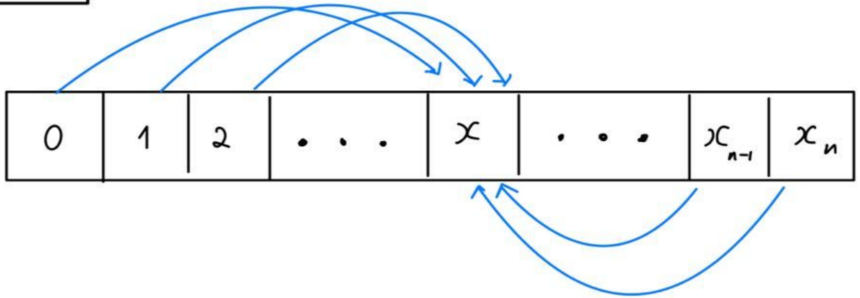
Advanced Models - Long-Short Term Memory Layer

- LSTM: Effective in remembering important information from earlier parts of the sequence and use it to process later parts.
- Bi-Directional LSTM: not only it can learn from the earlier parts of the data, but also the later parts.

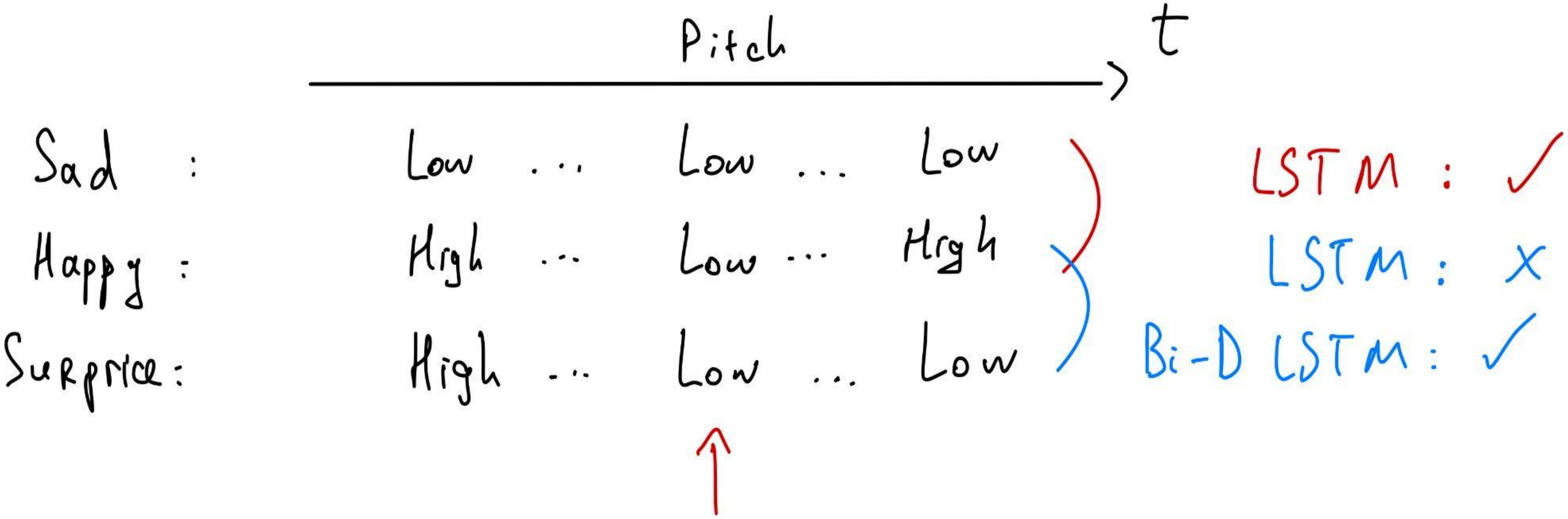


LSTM

Bi-Directional LSTM

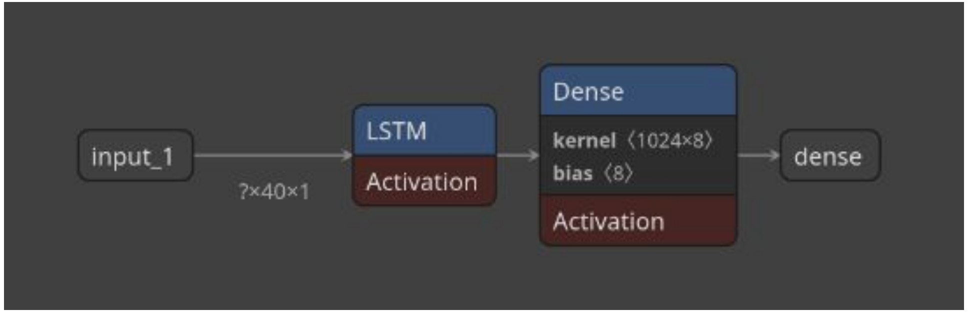


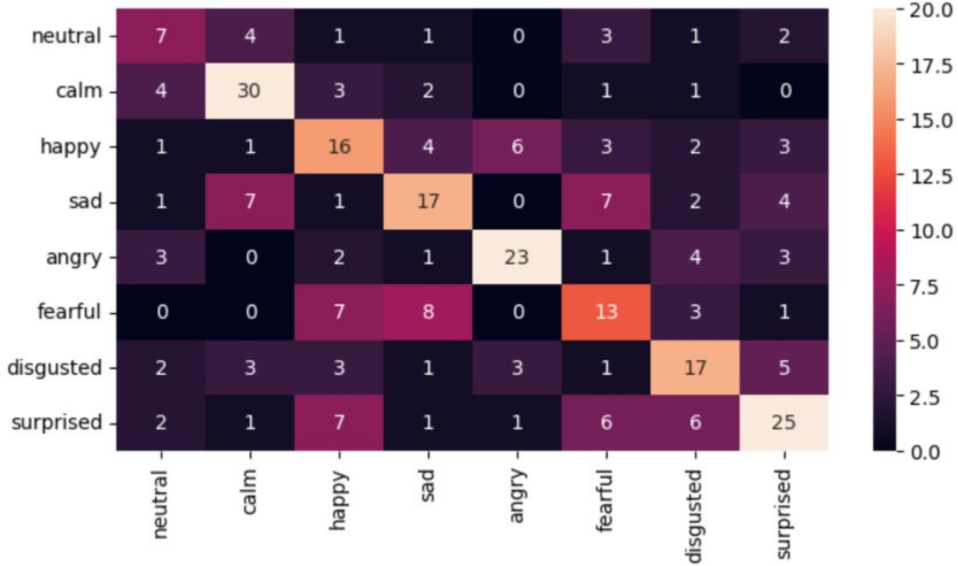
Advanced Models - Long-Short Term Memory Layer



Advanced Models - LSTM Network

- Model architecture: 1 layer of LSTM
- Parameters: 5,393,944
- Time to train: 2 minutes

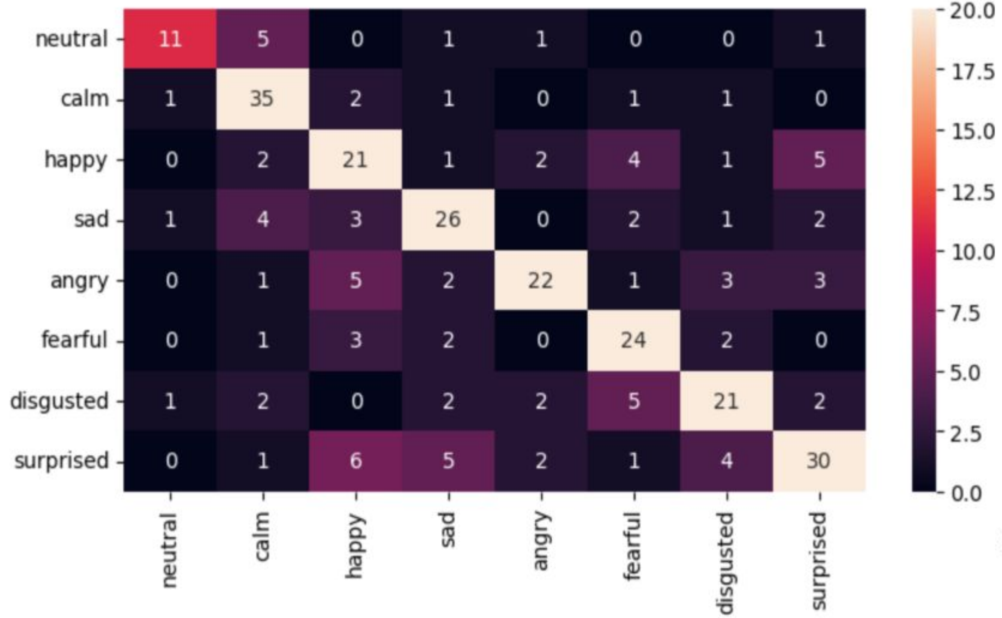




	precision	recall	f1-score	support
neutral	0.3500	0.3684	0.3590	19
calm	0.6522	0.7317	0.6897	41
happy	0.4000	0.4444	0.4211	36
sad	0.4857	0.4359	0.4595	39
angry	0.6970	0.6216	0.6571	37
fearful	0.3714	0.4062	0.3881	32
disgusted	0.4722	0.4857	0.4789	35
surprised	0.5814	0.5102	0.5435	49
accuracy			0.5139	288
macro avg	0.5012	0.5005	0.4996	288
weighted avg	0.5188	0.5139	0.5149	288

Advanced Models - CNN + LSTM Network

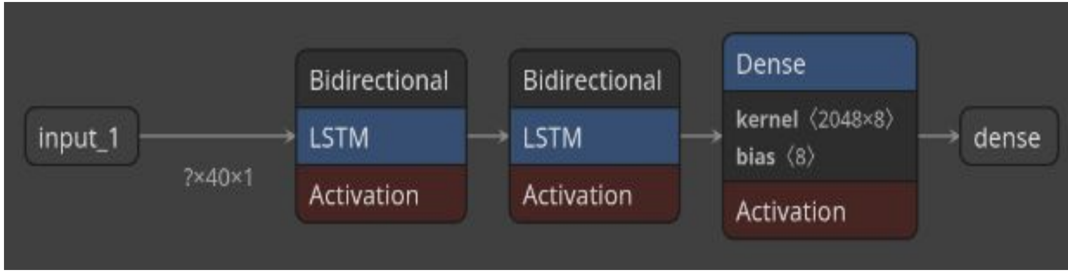
- Model architecture: Conv1D -> LSTM
- Parameters: 4,210,696
- Time to train: 1.5 minutes

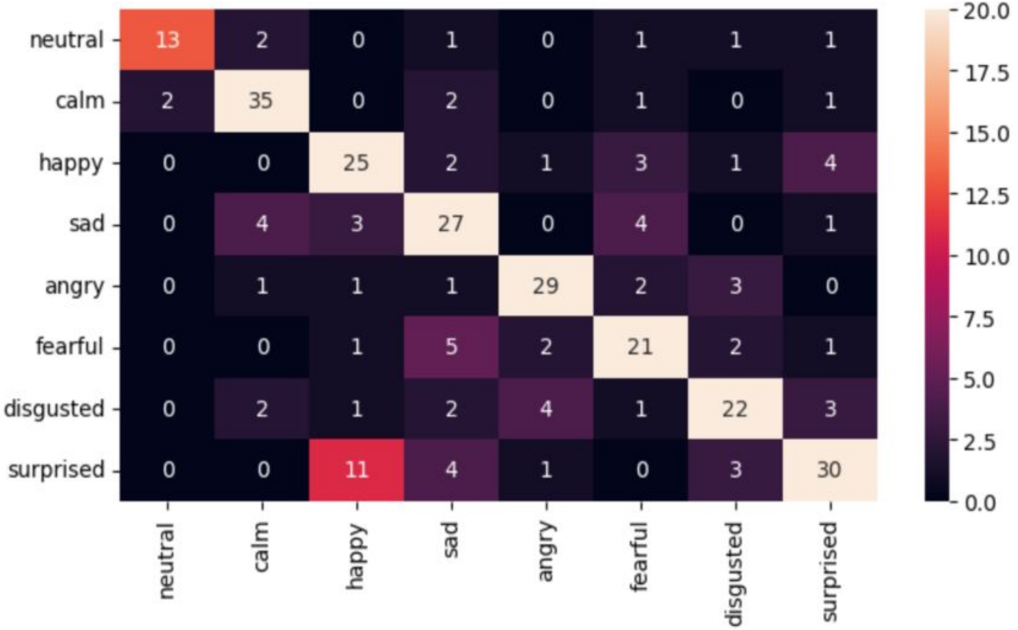


	precision	recall	f1-score	support
neutral	0.7857	0.5789	0.6667	19
calm	0.6863	0.8537	0.7609	41
happy	0.5250	0.5833	0.5526	36
sad	0.6500	0.6667	0.6582	39
angry	0.7586	0.5946	0.6667	37
fearful	0.6316	0.7500	0.6857	32
disgusted	0.6364	0.6000	0.6176	35
surprised	0.6977	0.6122	0.6522	49
accuracy			0.6597	288
macro avg	0.6714	0.6549	0.6576	288
weighted avg	0.6669	0.6597	0.6584	288

Advanced Models - Bi-directional LSTM Network

- Model architecture: 2 layer of Bi-Directional LSTM
- Parameters: 33,595,400
- Time to train: 13 minutes

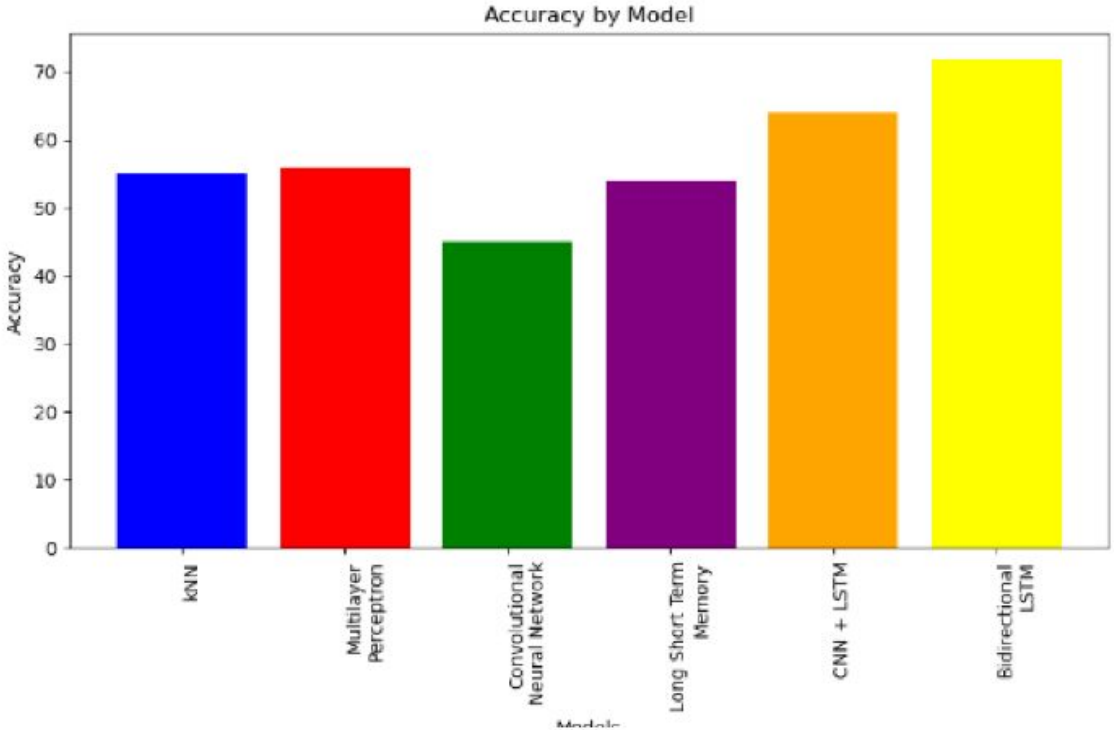




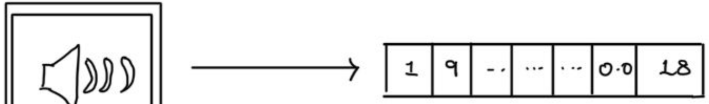
	precision	recall	f1-score	support
neutral	0.8667	0.6842	0.7647	19
calm	0.7955	0.8537	0.8235	41
happy	0.5952	0.6944	0.6410	36
sad	0.6136	0.6923	0.6506	39
angry	0.7838	0.7838	0.7838	37
fearful	0.6364	0.6562	0.6462	32
disgusted	0.6875	0.6286	0.6567	35
surprised	0.7317	0.6122	0.6667	49
accuracy			0.7014	288
macro avg	0.7138	0.7007	0.7041	288
weighted avg	0.7074	0.7014	0.7016	288

EVALUATION

Model Comparison



Why?



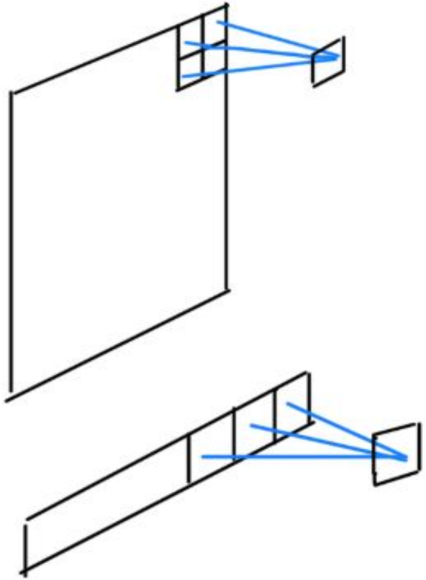
foo.wav

an array of MFCCs.

- One-dimensional
- Sequential
- Designed so if two pieces of audio sound similar to a human, they are close on the Mel scale.

CNN Performance

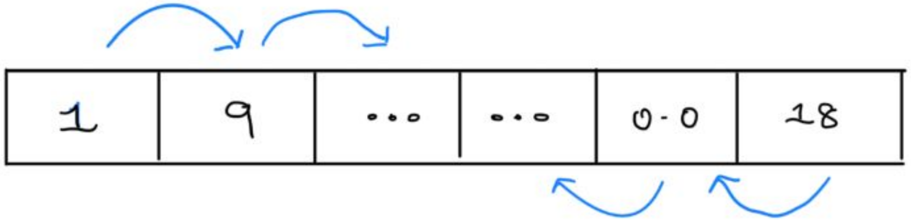
Worse than kNN, worse than MLP.



An array of MFCCs, while sequential, lack the strong local dependencies and spatial hierarchies CNNs typically exploit.

LSTM Performance

The best.



But LSTMs are RNNs and are designed specifically for <u>sequential</u> data.

What emotion was easiest to recognise?

Happy

Sad

Angry

Fearful

Calm

Neutral

Surprised

Disgusted

What emotion was easiest to recognise?

Happy

Sad

Angry

Fearful

Calm ← 87.89% accuracy

Neutral

Surprised

Disgusted ← 57.14% accuracy

Project hosted at:

https://github.com/sauryanshu55/Speech-Recognition/