Classification for Basic Voice Commands

HOW CAN DEEP LEARNING IMPACT SPEECH RECOGNITION ?

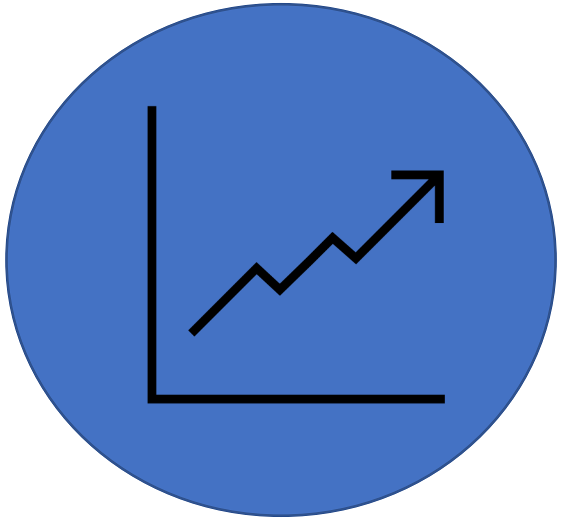
Maria Blinchevskaya
Raphael Delouya
Gadi G. Ezer
Claudia Palierne

1. INTRODUCTION

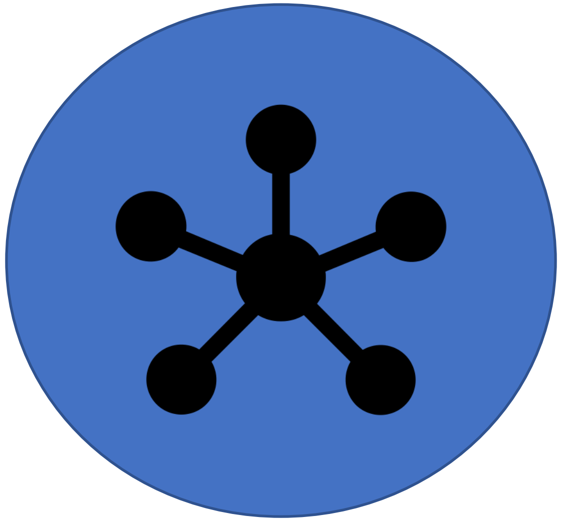
ORIGINAL MESSAGE :

Have you tried scaling your data using a MinMaxScaler ?

2. PROBLEM STATEMENT : Context



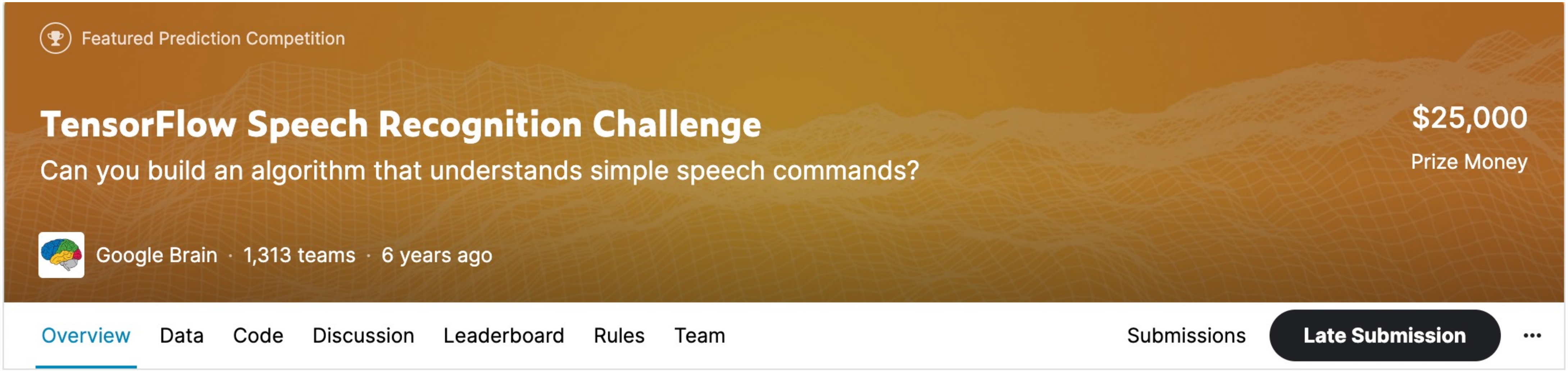
Rising demand
for smart devices
and voice-controlled
applications



Need for **efficient** and **accurate**
speech recognition technology

3. OUR PROJECT

Kaggle Competition (ended Jan - 2018)

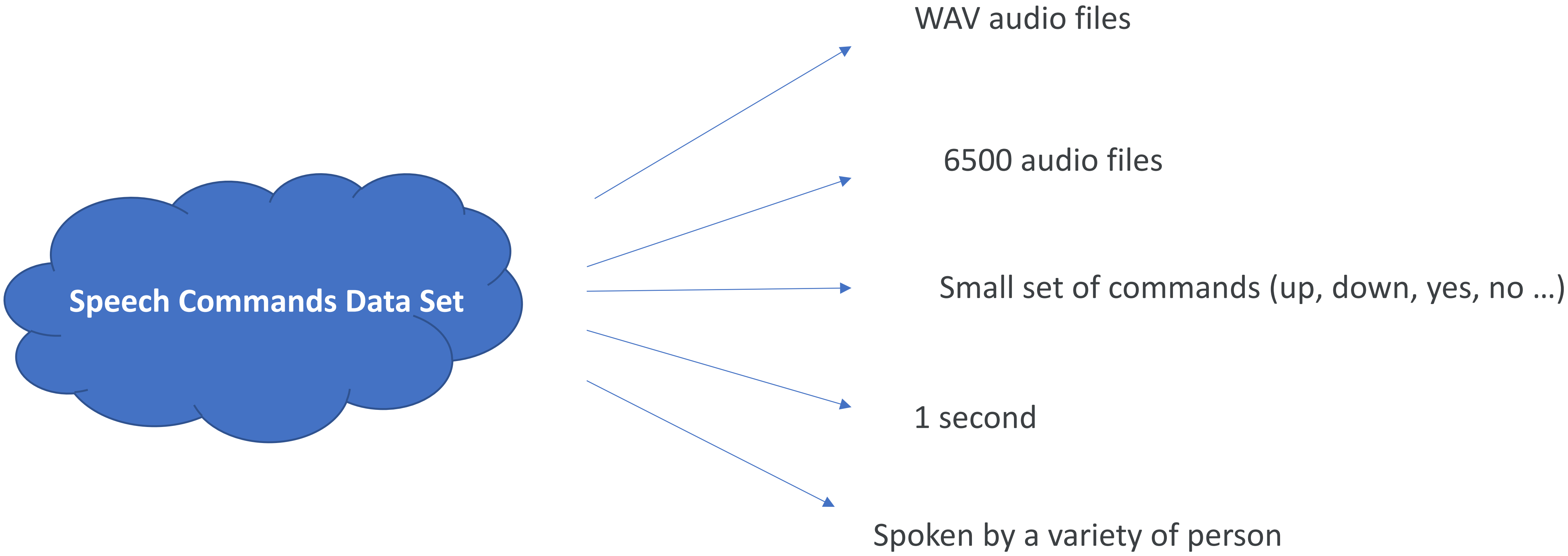


3. OUR PROJECT: Goals

Accurately identify and transcribe a range of basic words

Operate effectively in noisy environments

4. DATA COLLECTION AND PREPROCESSING



4. DATA COLLECTION AND PREPROCESSING

13 commands
(yes, no, one, two, three, four, five, six, seven, eight, nine, up, down)

↓

500 audio files for each command

↓

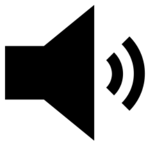
Added Background Noise

4. DATA COLLECTION AND PREPROCESSING: RAW AUDIO

STOP Audio File

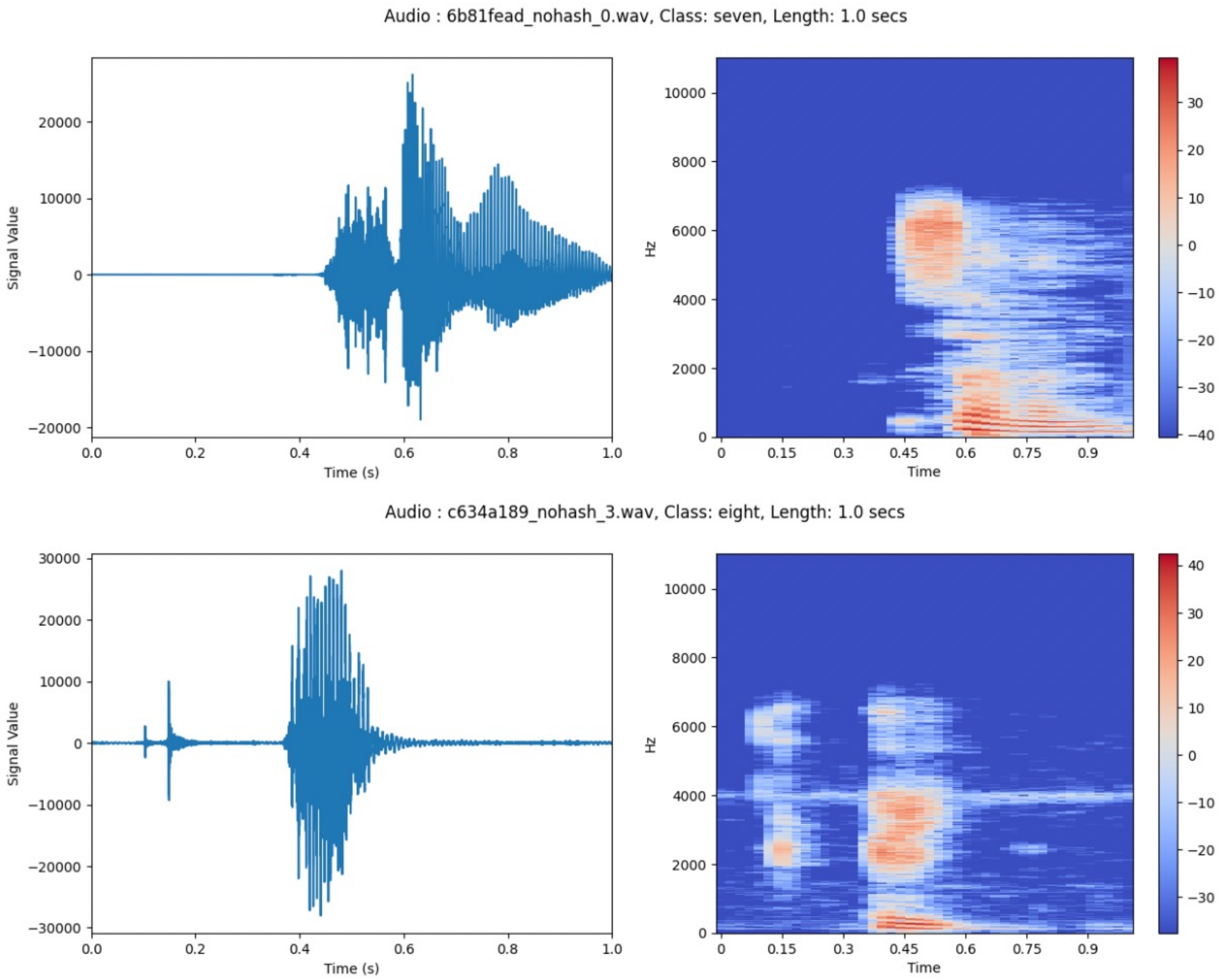
TWO Audio File

UP Audio File



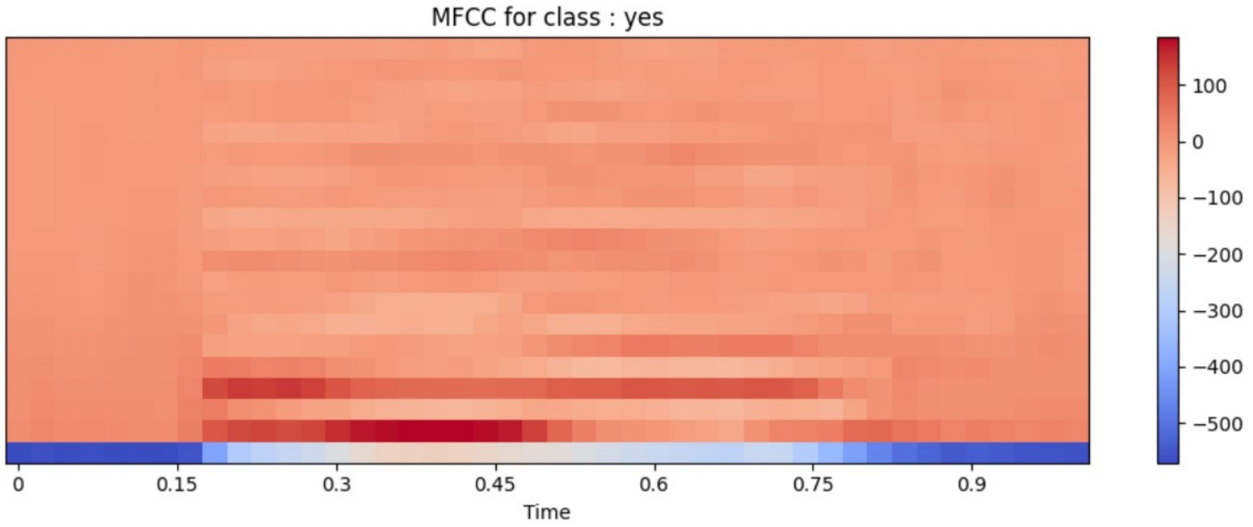
4. DATA COLLECTION AND PREPROCESSING: Waveform and Spectogram

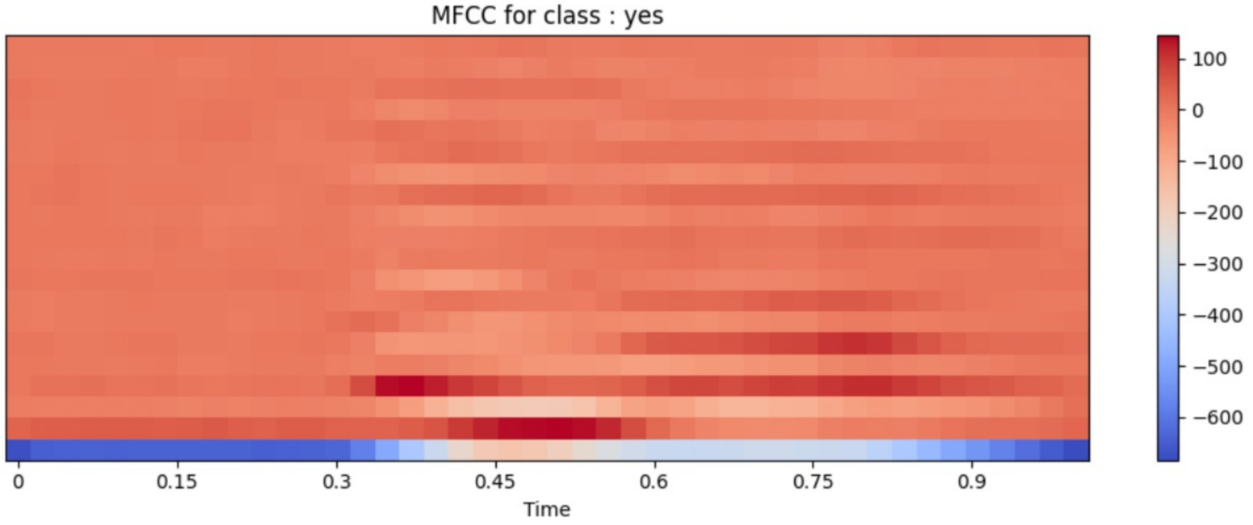
STFT (Short-Time Fourier Transform)



4. DATA COLLECTION AND PREPROCESSING: MFCC

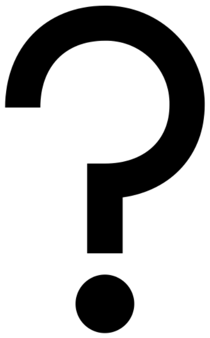
MFCC (Mel Frequency Cepstral Coefficients)





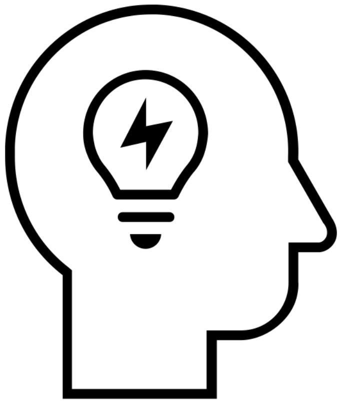
4. DATA COLLECTION AND PREPROCESSING: Challenges

PROBLEM



Initial Pre-processing

SOLUTION



LIBROSA library

5. MODEL DEVELOPMENT: TRAINING PROCESS

GENERAL INFO:

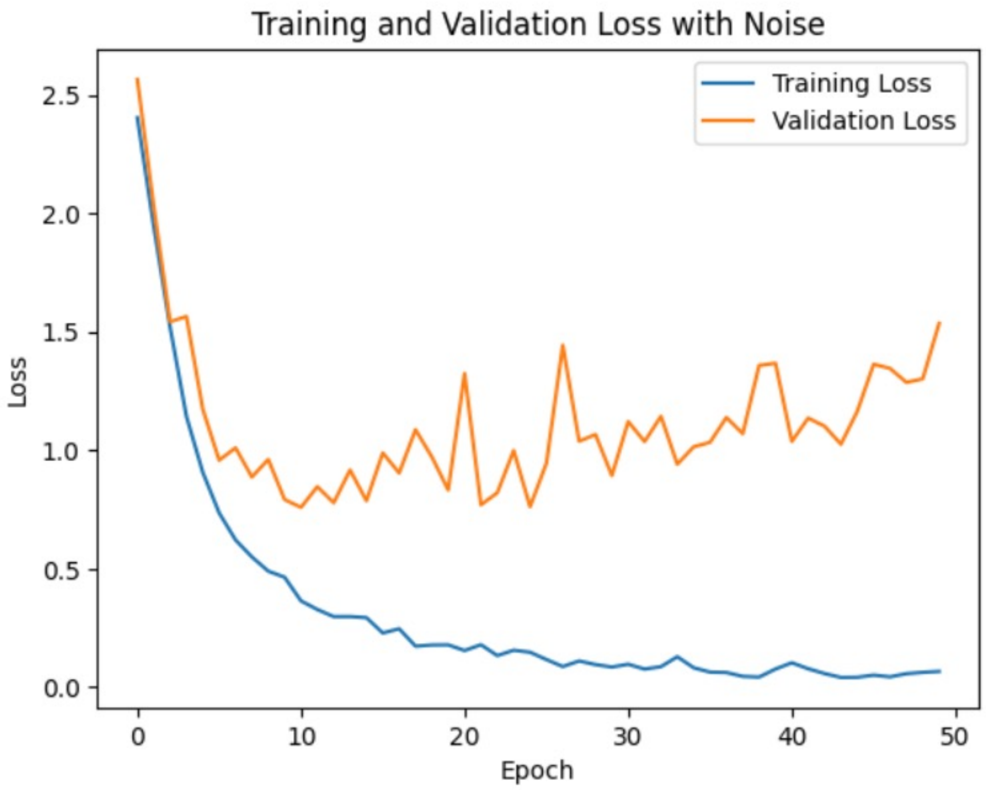
- Training Set, Validation Set, Test Set
- 50 epochs
- Accuracy Metric
- Adam Optimizer
- Learning Rate: 0.001

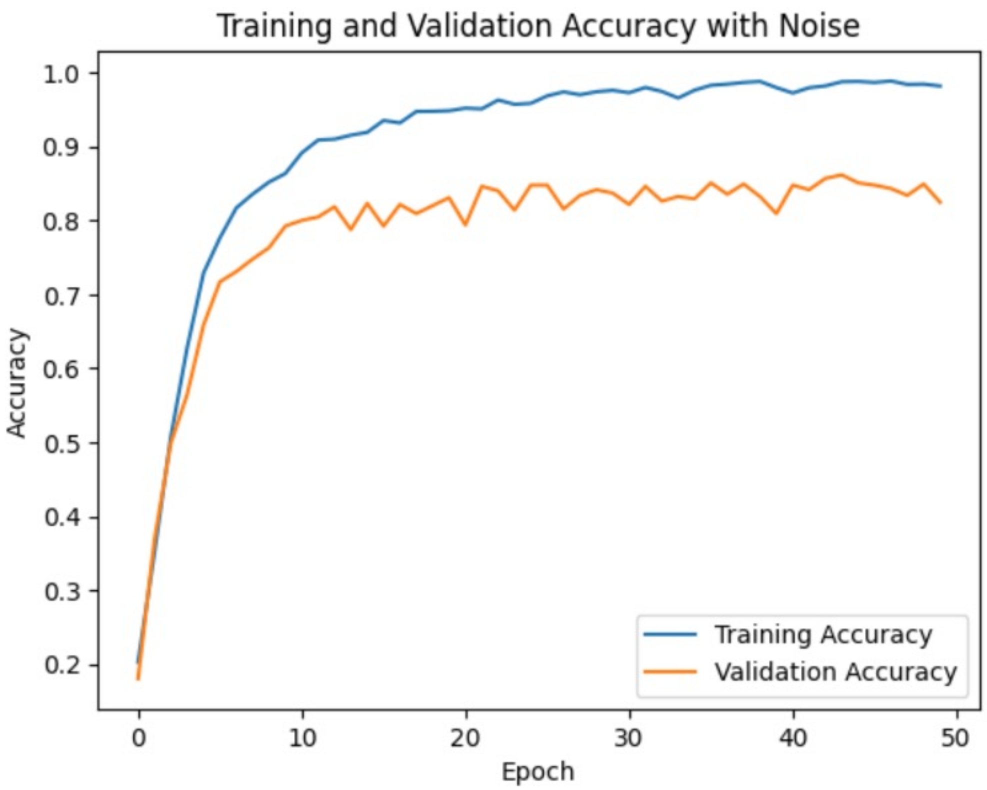
OUR MODEL:

- Sequential model architecture
- Reshape, Conv2D, MaxPooling2D, Flatten, Dense, and Dropout layers
- Smaller number of layers compared to AlexNet
- Fewer trainable parameters compared to AlexNet

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape_4 (Reshape)         (None, 20, 32, 1)         0

 conv2d_12 (Conv2D)          (None, 18, 30, 32)        320

 max_pooling2d_9 (MaxPooling  (None, 9, 15, 32)        0
 2D)

 conv2d_13 (Conv2D)          (None, 7, 13, 64)         18496

 max_pooling2d_10 (MaxPoolin  (None, 3, 6, 64)         0
 g2D)

 conv2d_14 (Conv2D)          (None, 1, 4, 128)         73856

 flatten_4 (Flatten)         (None, 512)               0

 dense_12 (Dense)            (None, 128)               65664

 dropout_8 (Dropout)         (None, 128)               0

 dense_13 (Dense)            (None, 64)                8256

 dropout_9 (Dropout)         (None, 64)                0

 dense_14 (Dense)            (None, 13)                845

=================================================================
Total params: 167,437
Trainable params: 167,437
Non-trainable params: 0
_________________________________________________________________
```

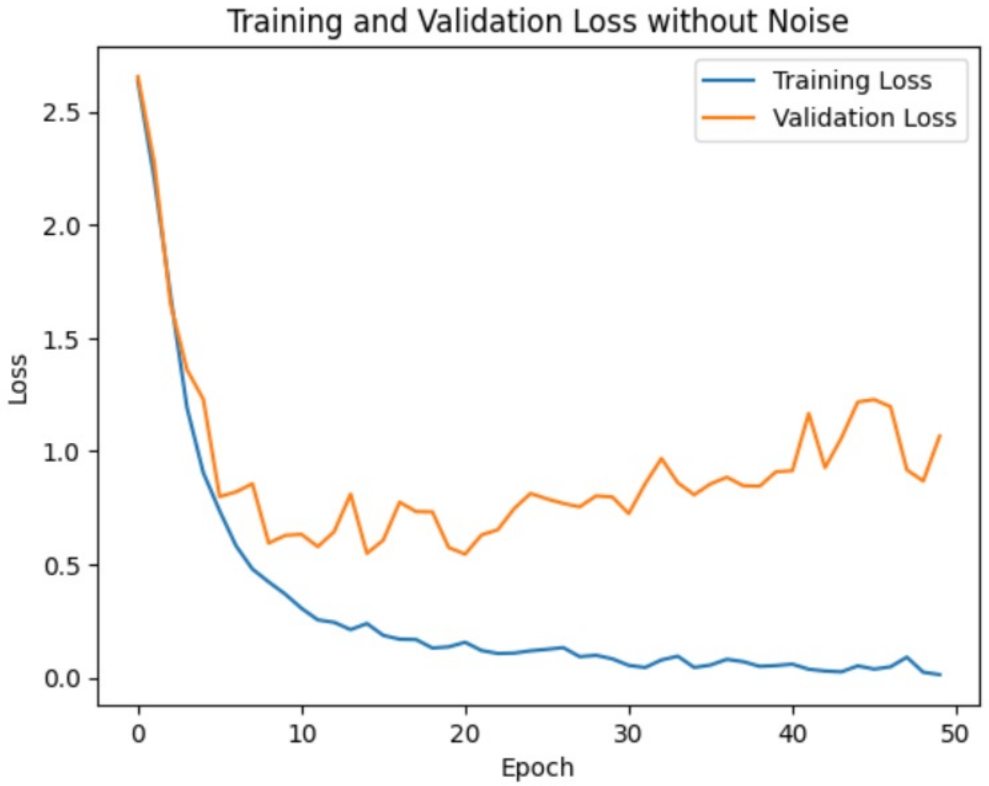
6. RESULTS AND EVALUATION : MFCC with noise

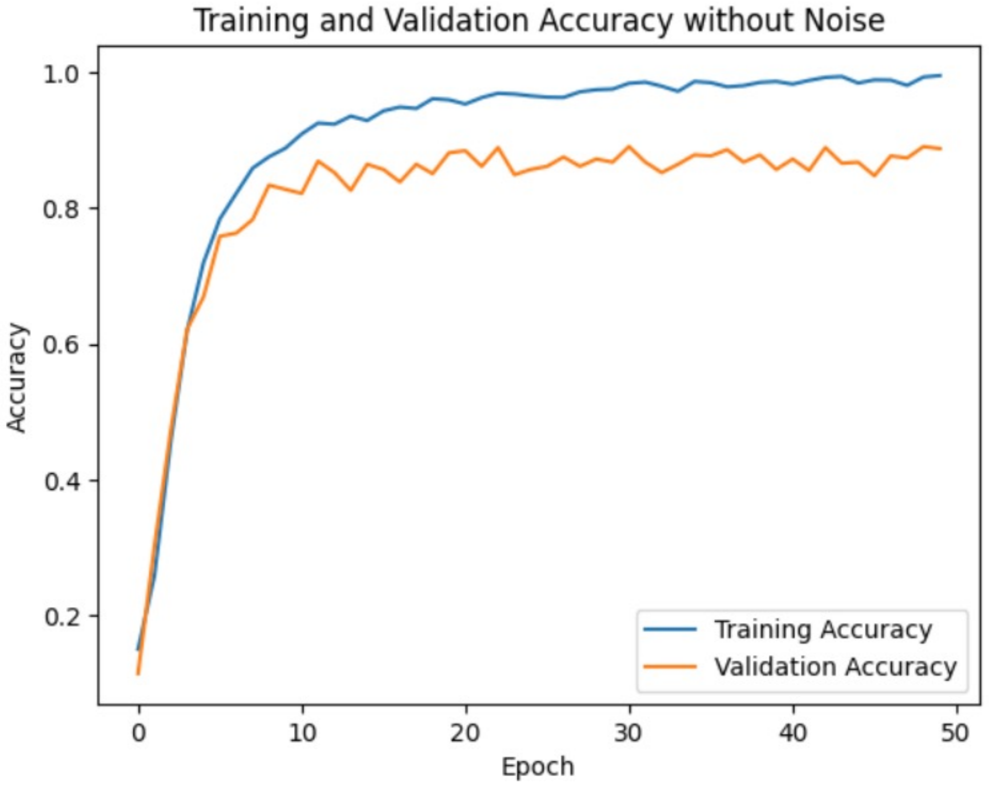




Validation Accuracy achieved: 0.86

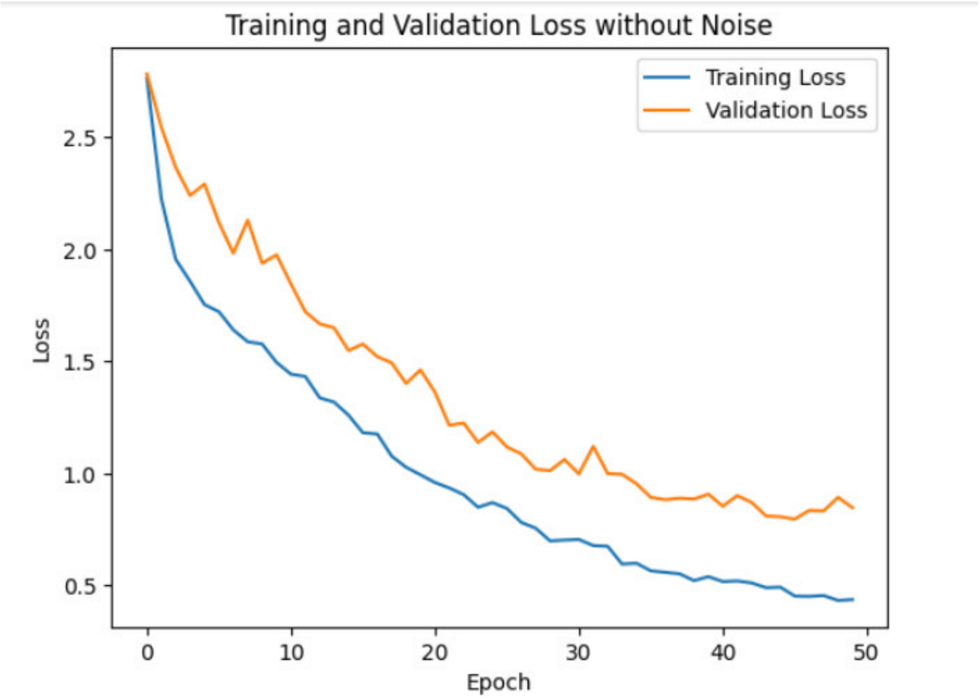
6. RESULTS AND EVALUATION : MFCC without noise

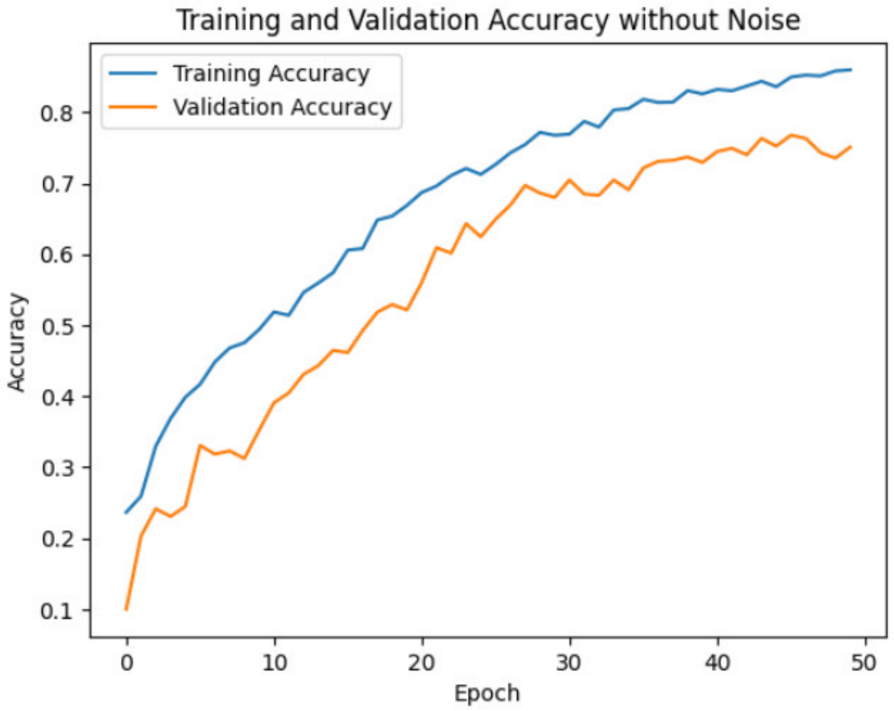




Validation Accuracy achieved: 0.88

6. RESULTS AND EVALUATION : Spectograms





Validation Accuracy achieved: 0.70

7. DEMO

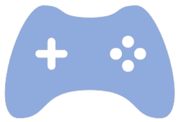
http://172.16.0.57:8501/

8. REAL-WORLD APPLICATIONS

VOICE
ASSISTANT

SMART
HOME

CARS



GAMING

9. CONCLUSION

RESULTS

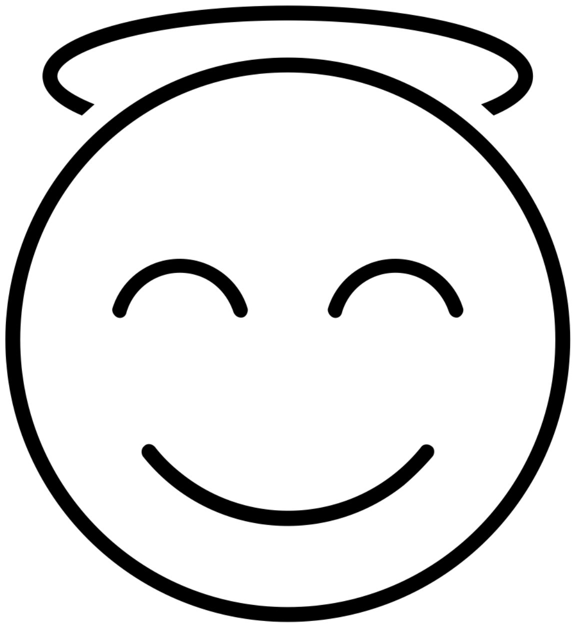
- Achieved very good accuracy with noise BUT overfits …
- Very inspiring project !

NEXT STEPS

- Increase the different commands
- Try with sentences
- Increase background noise

Last Word !

- Wav2Vec (META)



ANNEXES

MODEL DEVELOPMENT: WAV2VEC

- Deep learning model for speech recognition and speech-related tasks

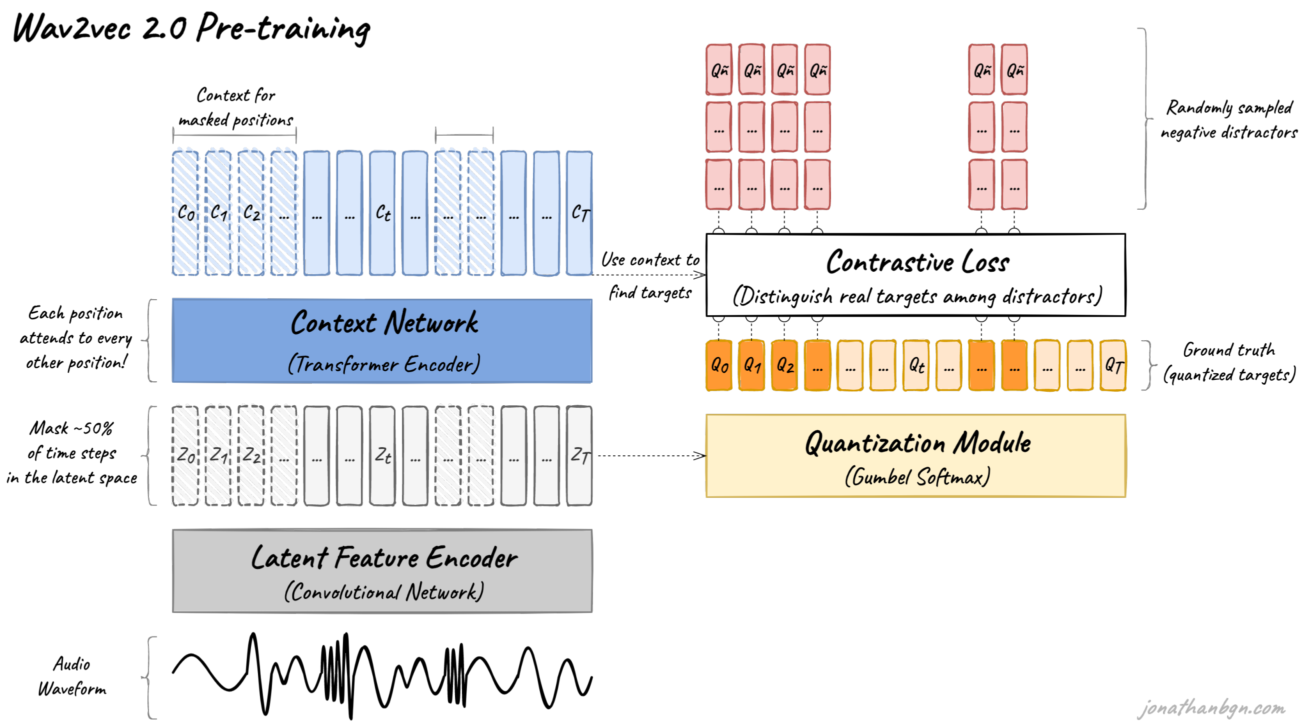
- State-of-the-art results in speech recognition benchmarks

- Handles raw audio data directly, no manual feature extraction needed

- Uses transformers to process CNN output for feature extraction

- Transformer models capture long-range dependencies and contextual information in audio sequences

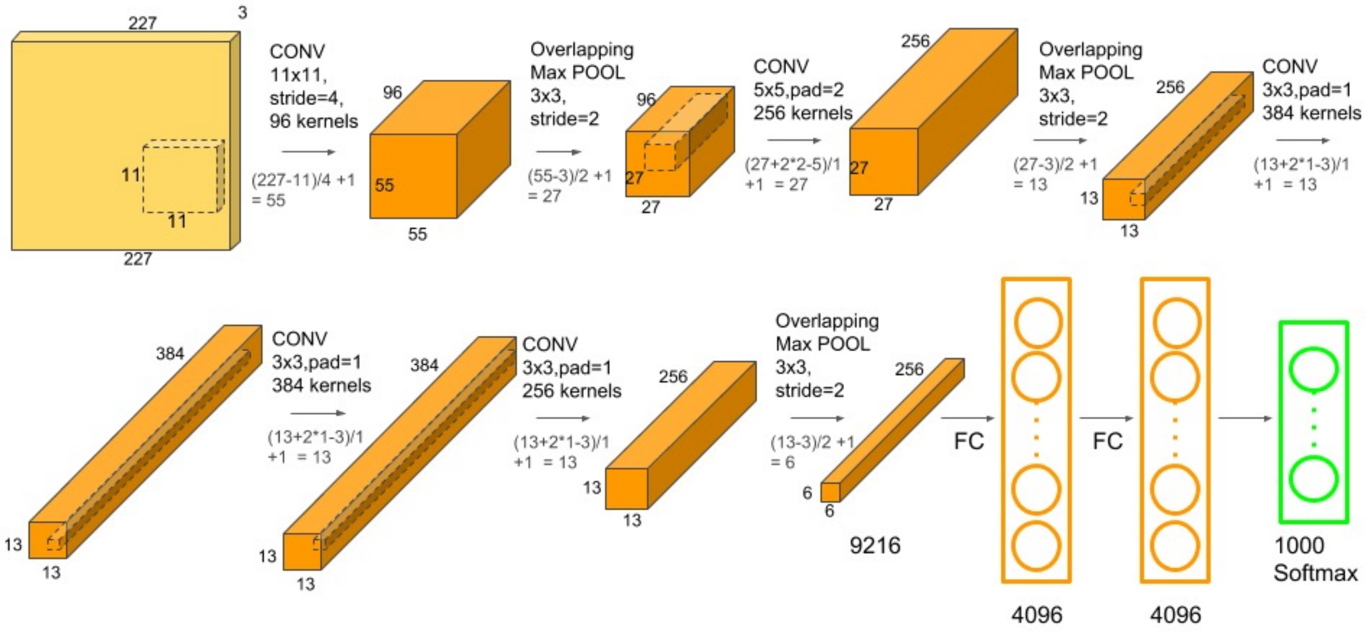
•Pre-trained: 72% results



KEY METRIC: Levenshtein

MODEL DEVELOPMENT: AlexNet CNN

- Deep convolutional neural network architecture
- Multiple layers: convolutional, max-pooling, and fully connected
- Eight layers in total, with the first five being convolutional
- Convolutional layers extract low-level features
- Max-pooling layers downsample feature maps
- Fully connected layers serve as classifier
- ReLU activation functions used
- Dropout regularization implemented



KEY METRIC: Accuracy