Speech Command Recognition

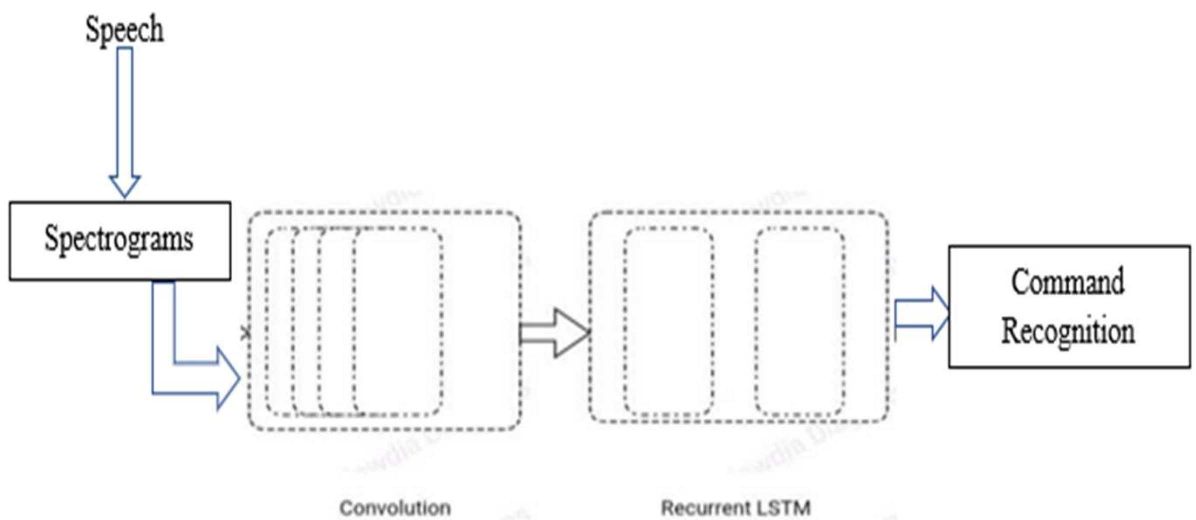
Made By :
Adina Dingankar
Rehapriadarsini Manikandasamy

What is Speech Command Recognition?

Speech command recognition has extensive real-world applications to identify the import words/phrases in the spoken sentences.

One such application is Keyword spotting (KWS) technology, a potential technique to provide fully hands-free interface, and this is especially convenient for mobile devices compared to typing by hands.

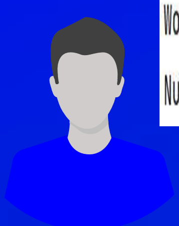
Implementation of keyword spotting technology involves networks which helps in understanding the specific commands in the conversations.



Source: [(PDF) A Hybrid Technique using CNN+LSTM for Speech Emotion Recognition (researchgate.net)]

Dataset Description

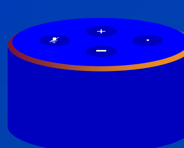
The dataset used in this project is sourced from Kaggle which has audio files (.wav) of duration of one second. The dataset has approximately 30K audio files classified based on 14 major labels. The dataset will be separated into train-test sets manually and processed to be trained on the deep learning models.



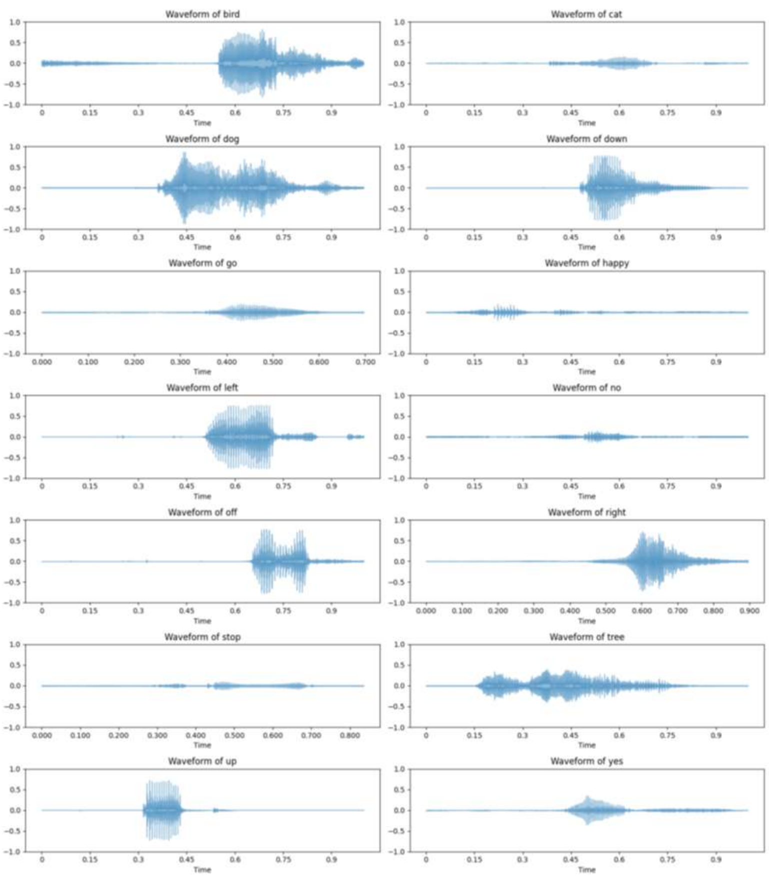
Words spoken: ['left', 'right', 'no', 'happy', 'dog', 'go', 'stop', 'bird', 'yes', 'off', 'down', 'up', 'cat', 'tree']

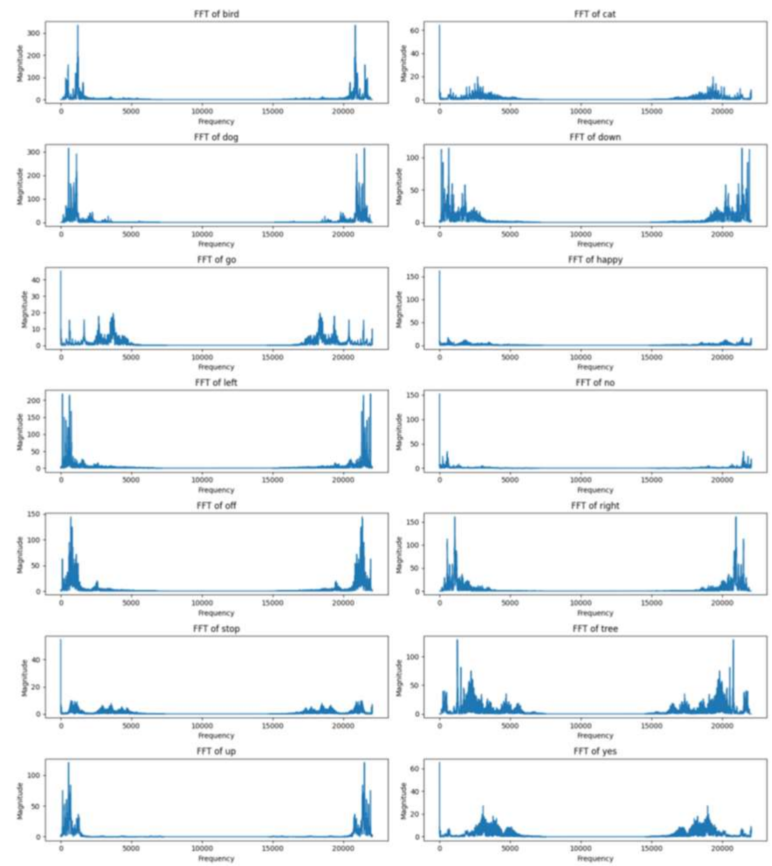
Number of files total: 29690

```
data = {
    "mapping": [],
    "labels": [],
    "MFCCs": [],
    "files": []
}
```

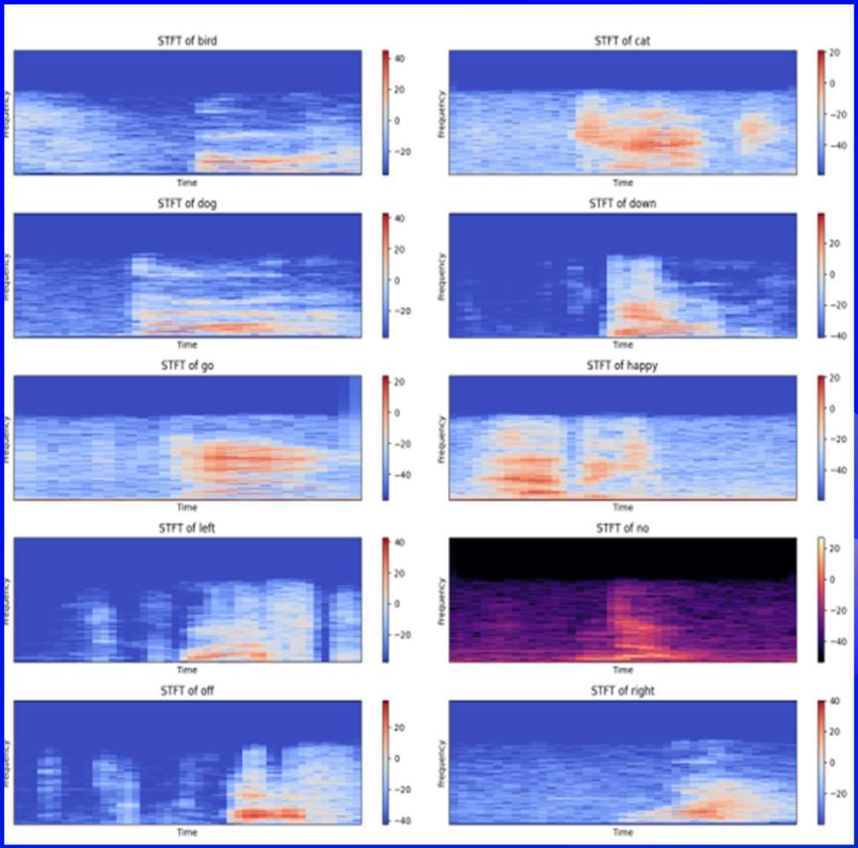


Waveforms and FFT of each Sub Category

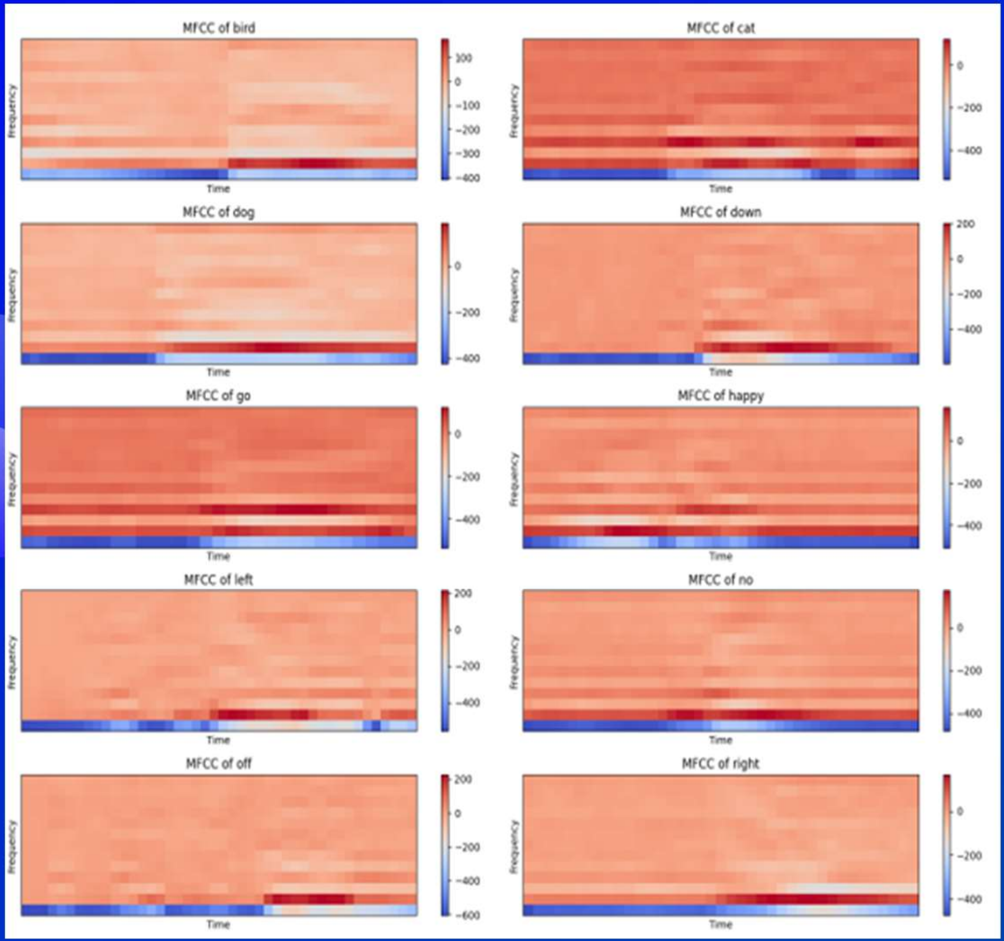




MFCC



STFT of bird
STFT of cat
STFT of dog
STFT of down
STFT of go
STFT of happy
STFT of left
STFT of no
STFT of off
STFT of right
Time
Frequency



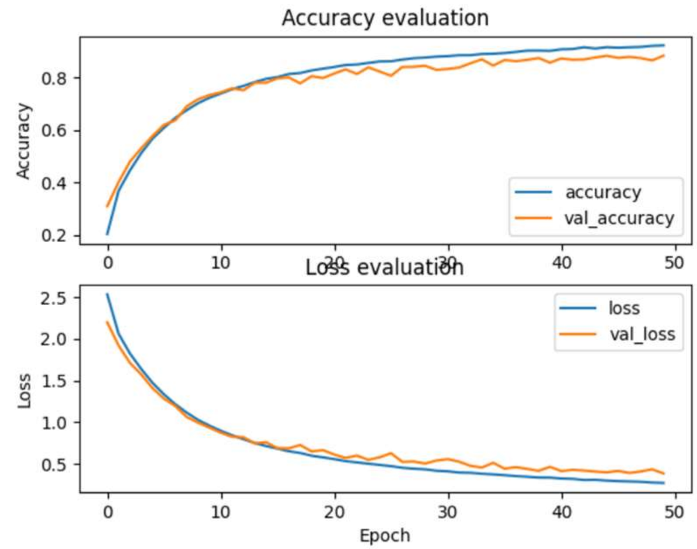
MFCC of bird
MFCC of cat
MFCC of dog
MFCC of down
MFCC of go
MFCC of happy
MFCC of left
MFCC of no
MFCC of off
MFCC of right
Time
Frequency

STFT

CNN

CNN model on training on 50 epochs resulted in accuracy of 87% on the test set. The accuracy and loss evaluation plots shows that the model has been training better on the train data as we can see a smooth curve. When it was validated based on validation set there has been a fluctuations in the loss and accuracy values which shows the model might be overseeing the data.

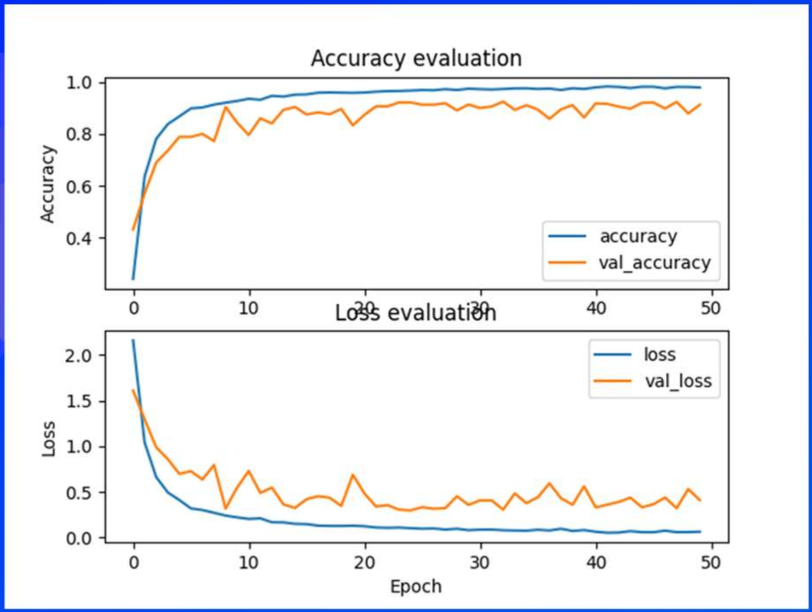
```
250/250 [==============================] - 1s 2ms/step - loss: 0.3955 - accuracy: 0.8791

Test loss: 0.39549198746681213, test accuracy: 87.9107117652893
```



CNN + LSTM

A convolutional neural network being a feed-forward network, filters spatial data whereas the recurrent neural network (LSTM) feeds data back into itself. Thus, recurrent neural networks are better suited for sequential data. A convolutional neural network can perceive patterns across space, LSTM can see them over time. Since speech signal is sequential, so LSTM is best suited for speech processing.

```
204/204 [==============================] - 2s 12ms/step - loss: 0.0768 - accuracy: 0.9746 - val_loss: 0.4394 - val_accuracy: 0.8972
Epoch 48/50
204/204 [==============================] - 2s 12ms/step - loss: 0.0602 - accuracy: 0.9803 - val_loss: 0.3220 - val_accuracy: 0.9223
Epoch 49/50
204/204 [==============================] - 2s 12ms/step - loss: 0.0608 - accuracy: 0.9798 - val_loss: 0.5318 - val_accuracy: 0.8780
Epoch 50/50
204/204 [==============================] - 2s 12ms/step - loss: 0.0650 - accuracy: 0.9779 - val_loss: 0.4105 - val_accuracy: 0.9115


250/250 [==============================] - 1s 3ms/step - loss: 0.4082 - accuracy: 0.9057


Test loss: 0.4082329571247101, test accuracy: 90.5693531036377


Process finished with exit code 0
```

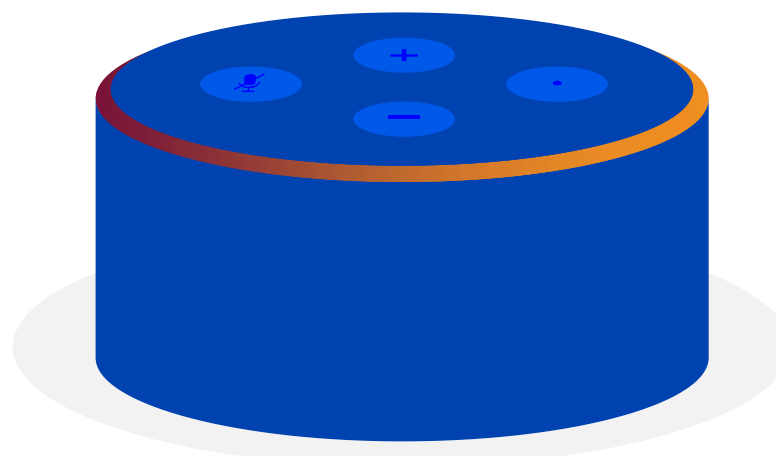


CNN + LSTM

```
=================================================================
conv2d (Conv2D)             (None, 44, 13, 128)       1280

batch_normalization (BatchN  (None, 44, 13, 128)      512
ormalization)

activation (Activation)     (None, 44, 13, 128)       0

max_pooling2d (MaxPooling2D  (None, 22, 7, 128)       0
)

conv2d_1 (Conv2D)           (None, 22, 7, 72)         83016

batch_normalization_1 (Batc  (None, 22, 7, 72)        288
hNormalization)

activation_1 (Activation)   (None, 22, 7, 72)         0

max_pooling2d_1 (MaxPooling  (None, 6, 2, 72)         0
2D)

conv2d_2 (Conv2D)           (None, 6, 2, 64)          41536

batch_normalization_2 (Batc  (None, 6, 2, 64)         256
hNormalization)

activation_2 (Activation)   (None, 6, 2, 64)          0

max_pooling2d_2 (MaxPooling  (None, 2, 1, 64)         0
2D)

conv2d_3 (Conv2D)           (None, 2, 1, 64)          36928

batch_normalization_3 (Batc  (None, 2, 1, 64)         256
hNormalization)

activation_3 (Activation)   (None, 2, 1, 64)          0

max_pooling2d_3 (MaxPooling  (None, 1, 1, 64)         0
2D)

reshape (Reshape)           (None, 1, 64)             0

lstm (LSTM)                 (None, 32)                12416

dense (Dense)               (None, 14)                462

=================================================================
```

Results

Model	Train accuracy	Train loss	Test accuracy	Test loss
CNN	0.91	0.30	0.87	0.39
CNN+LSTM	0.97	0.06	0.905	0.40



Future Scope:

The attempt on training WideResnet and VGG19 models failed. The future scope of the project can involve usage of deep pretrained models by handling the data preprocessing part much better. And finding a test data with significant modulation differences in the words uttered would help in finding a better and accurate model.