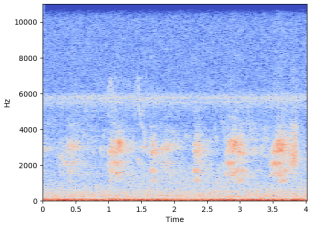
DL Assignment 2: Speech Classification

Anish Madan, Shagun Uppal, Sarthak Bhagat

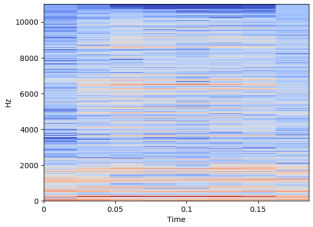
March 2019

1 Visualization with sample spectrograms of different classes

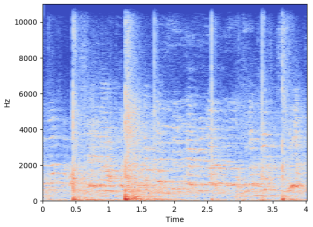
1.1 STFT Features Plots



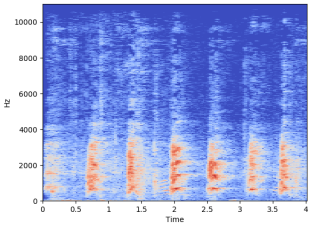
(a) Air Conditioner



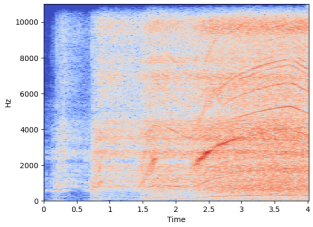
(b) Car Horn



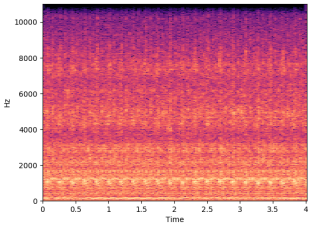
(c) Children Playing



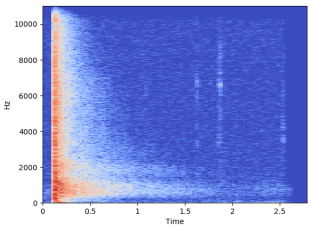
(d) Dog Bark



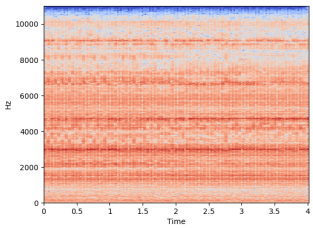
(e) Drilling



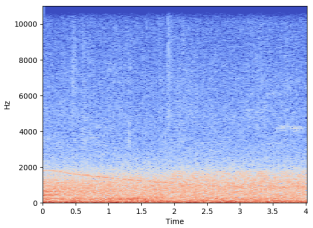
(f) Engine Idling



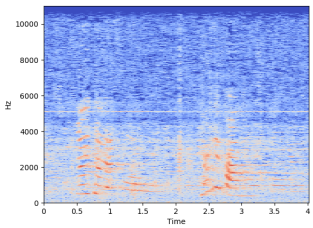
(g) Gun Shot



(h) Jack Hammer

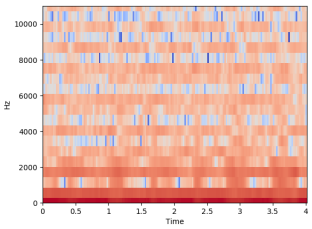


(i) Siren

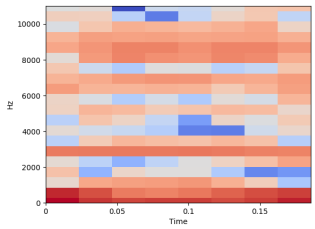


(j) Street Music

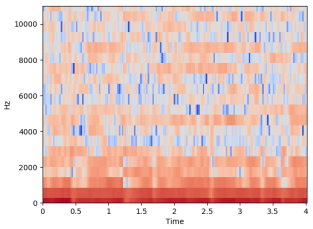
1.2 MFCC Feature Plots



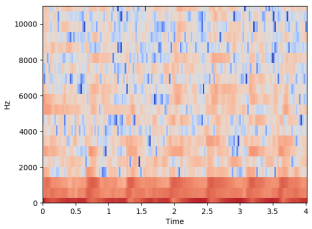
(a) Air Conditioner



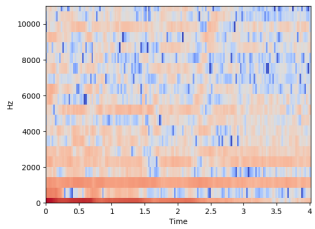
(b) Car Horn



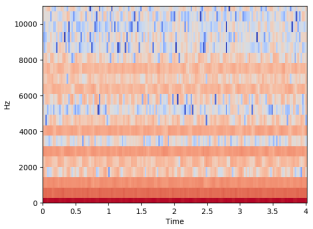
(c) Children Playing



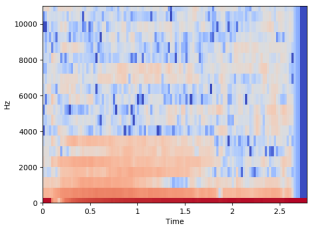
(d) Dog Bark



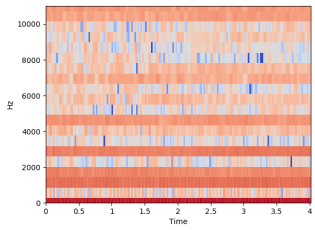
(e) Drilling



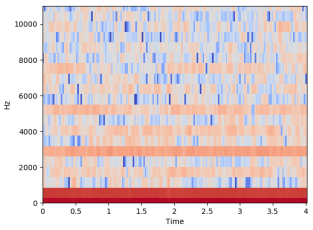
(f) Engine Idling



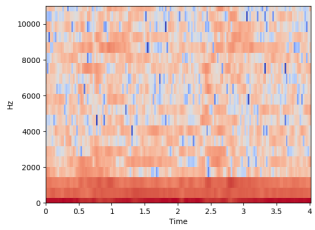
(g) Gun Shot



(h) Jack Hammer

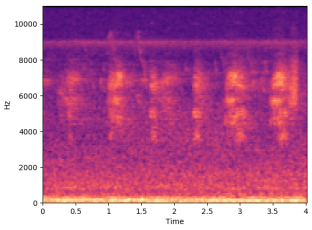


(i) Siren

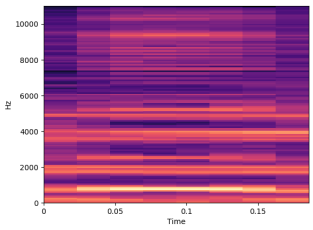


(j) Street Music

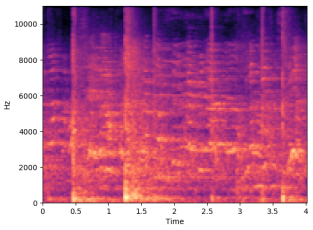
1.3 Spectrogram Feature Plots



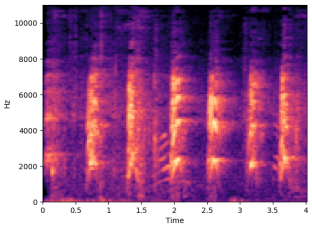
(a) Air Conditioner



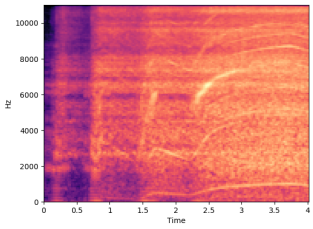
(b) Car Horn



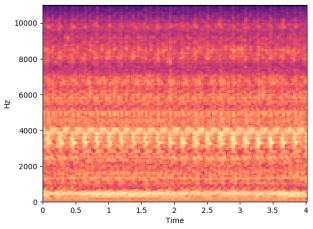
(c) Children Playing



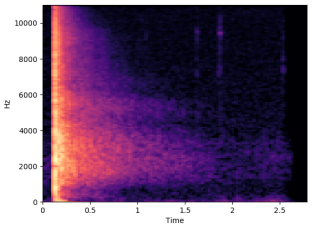
(d) Dog Bark



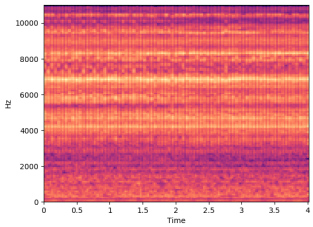
(e) Drilling



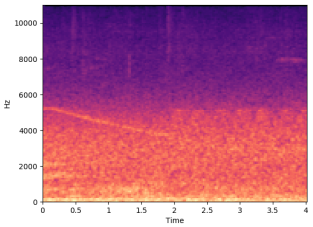
(f) Engine Idling



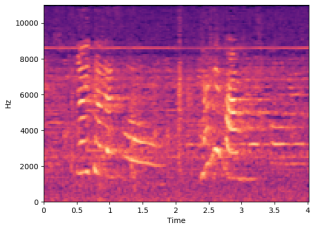
(g) Gun Shot



(h) Jack Hammer



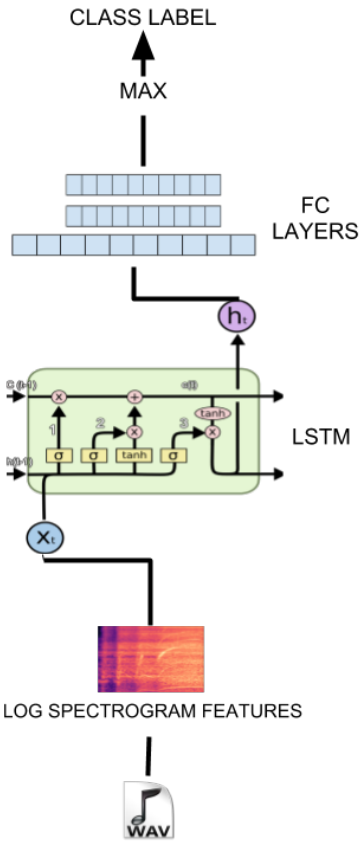
(i) Siren



(j) Street Music

2 Network Architecture

2.1 Block Diagram



(a) Block Diagram for our model

2.2 Motivation

The first design choice involved choosing an RNN variant. Out of the common RNN variants, we chose to use an LSTM since it overcomes the pitfalls of a vanilla RNN by modelling long-term dependencies better. In theory a GRU would have worked better by reducing overfitting in LSTM, but we did not achieve any stark difference between the results of LSTM and GRU, hence we chose to stick with LSTM.

We further connect the output of LSTM to Linear FC layers for classification, where the last layer comprises of 10 neurons corresponding to 10 classes. To reduce overfitting, we used dropout in LSTM as well as Linear Layers. The LSTM extracts the necessary sequencial features from the audios and feeds it to the fully connected layers that classify the audios based on extracted features into 10 classes.

3 Training Approach

3.1 Explanation for Training Approach

After preprocessing, the features extracted from each audio file are passed through the LSTM network with hidden size 5300 and number of layers 1. The LSTM processes sequential data of the audio and further passes it to the subsequent fully connected layers used for classification over the extracted features. The linear layers finally reduce the dimension of the LSTM output to the dimension equal to the number of classes i.e 10 for our dataset.

We used Cross Entropy Loss with Adam optimizer for training our network.

3.2 Data Preprocessing

The audio clips in this dataset had variable lengths and therefore, we transformed them to have equal lengths but repeating them to the closest ceil/greatest integer and then clipping it to become 4 seconds in length. This kind of a preprocessing was necessary as the features that we used were log spectrogram features which are time-domain features, hence, will be of different length for audio clips with different lengths. But as our network requires inputs of fixed sizes, we had to ensure they have the same length in order to feed it to our LSTM.

3.3 Libraries Used for extracting features

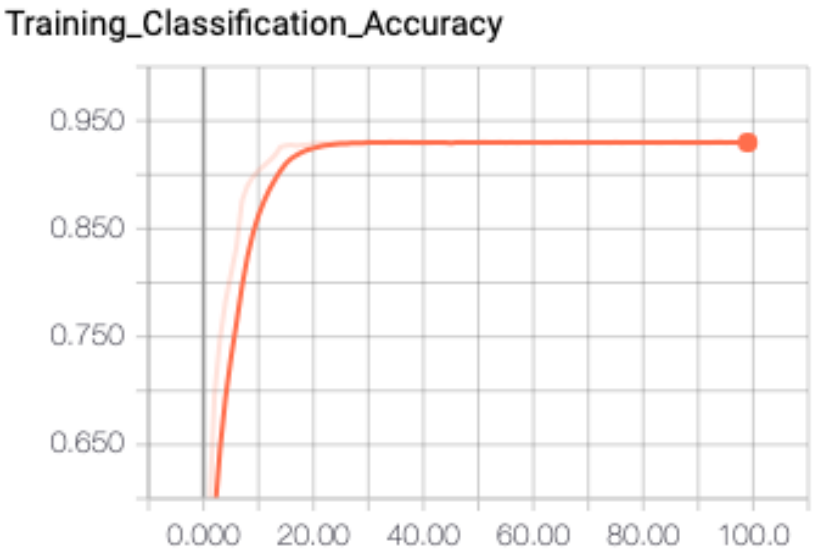
pydub: We used this for editing or preprocessing of the audio clip in order to make it fixed length.

librosa: This library has in-built function for extracting the amplitude of log spectrogram features for the audio clips. Along with the amplitude of log spectrograms as features, we also use delta information of log spectrograms as features since in audio, the order is important and by incorporating delta features, we are more attentive to change.

4 Loss Plots

4.1 Plots for Train Set

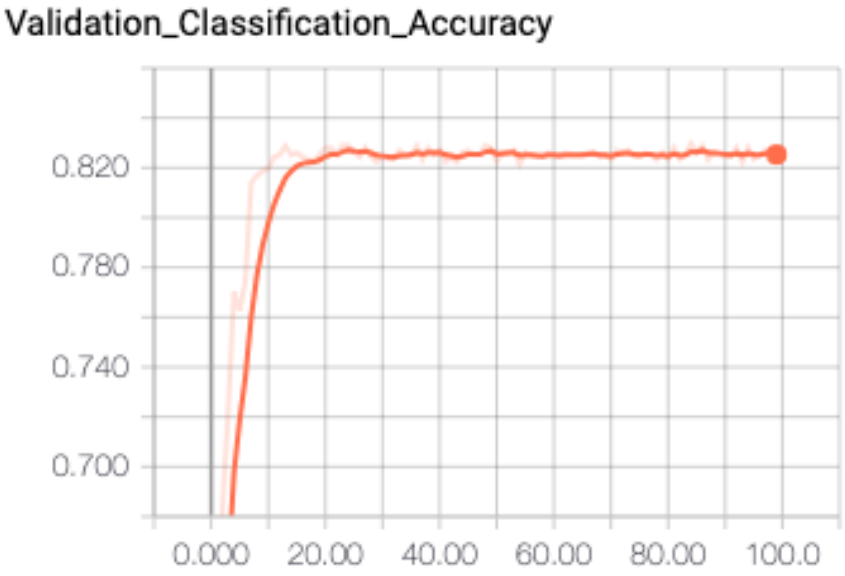
(a) Train Loss



(a) Train Accuracy

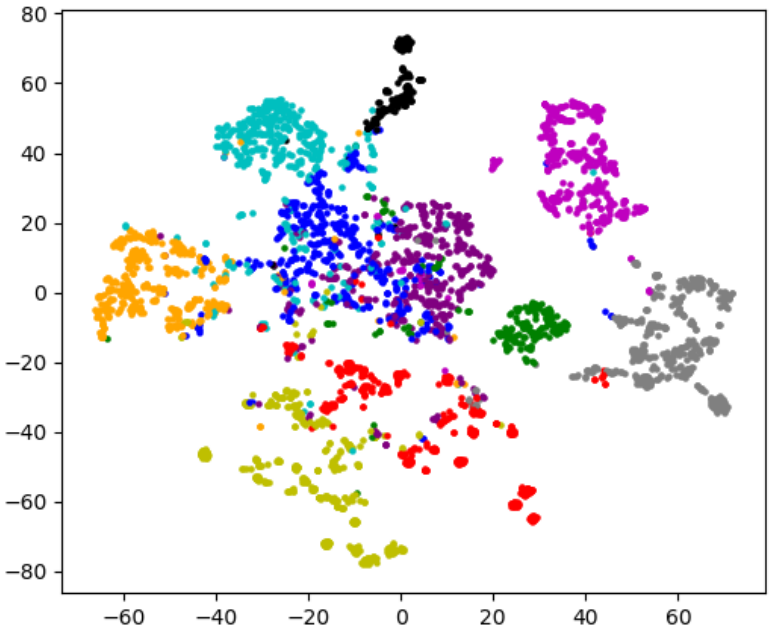
4.2 Plots for Validation Set

(a) Train Loss



(a) Train Accuracy

5 Visualization of last layer features



(a) TSNE visualization of features of last layer

6 Model Performance

6.0.1 Accuracy

Train Set: 93.75%

Validation Set: 82.98%

6.0.2 Loss

Train Set: 0.29

Validation Set: 0.47