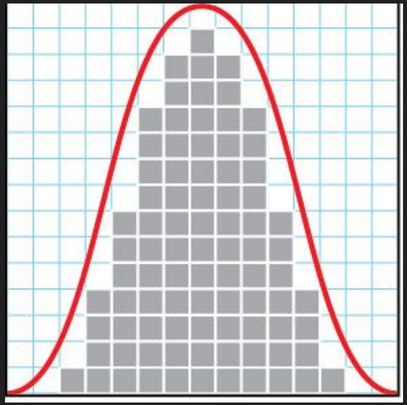
MACHINE LEARNING PROJECT
INTERIM MEETING

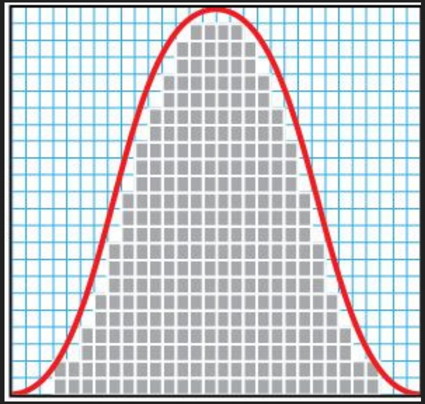
Audio Super Resolution

The aim is to achieve this via applying various Machine Learning models and comparing results



Input Signal
(Low Res)

Black Box
(model)



Output Signal
(High Res)

Team: Ramya YS (2015117)   Manik Arora (2015053)   Akarsha Sehwag (2015010)   Shiven Mian (2015094)

Dataset Used and Preprocessing

- Corpus Name: CSTR VCTK Dataset, Piano Dataset
- Corpus Size: 109 Native Speakers. Each speaker reads about 400 sentences (VCTK). Beethoven sonatas (Piano)

- Pre-processing →
  - We first extracted the audio files into floating point values
  - We created low and high resolution pairs of data, where low resolution data was used as input, and our aim is to resolve this input close to the high resolution data
  - Procedure: Sampling and padding

- Training Set: 88% of the sample space
- Validation Set: 6% of the sample space
- Test Set: 6% of the sample space

Design Choices

- Model Selection: We decided to try out both reconstruction (autoencoder) as well as generative approaches (LSTM, HMM, SRGAN (adaptation of Image SRGAN), ResNet (Kuleshov et al))
- Final models will be selected by the most accurate solution obtained, as defined by our loss function (Mean Squared Error Loss). We calculate this loss between our output signal and the high resolution ground truth signal.
- In our current approach, we have tested an Autoencoder-based reconstruction approach with the following hyperparameters:
    - Batch Size = 16 (vs 4 vs 1)
    - Epochs = 4 (vs 1)
    - Input -->Layer 1--(Activation: relu)--> Layer 2--(relu)--> Layer 3 (relu) → Output Layer
- Explanation for Hyperparameters Choice
    - Speed
    - Accuracy
    - Convergence

Challenges Faced Yet

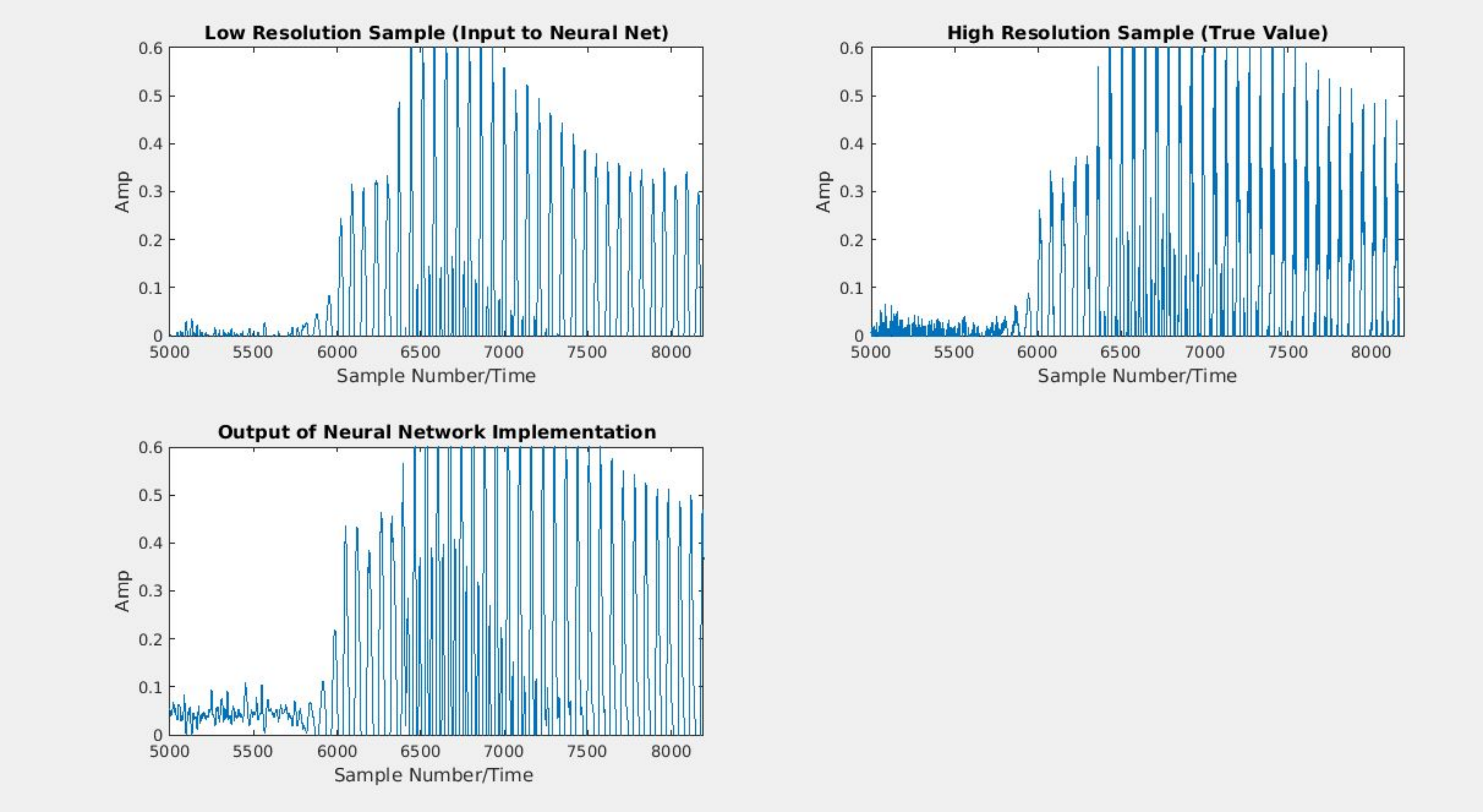
- Technical Challenges: Library issues in HPC, therefore we have currently used partial set of the entire data. (2432 audio samples of 3 seconds)
- Scaling and Shifts in Results
- Accuracy and performance of the trained models (autoencoder, as expected, didn't give us good results). Though we expect the generative models to give us better results.

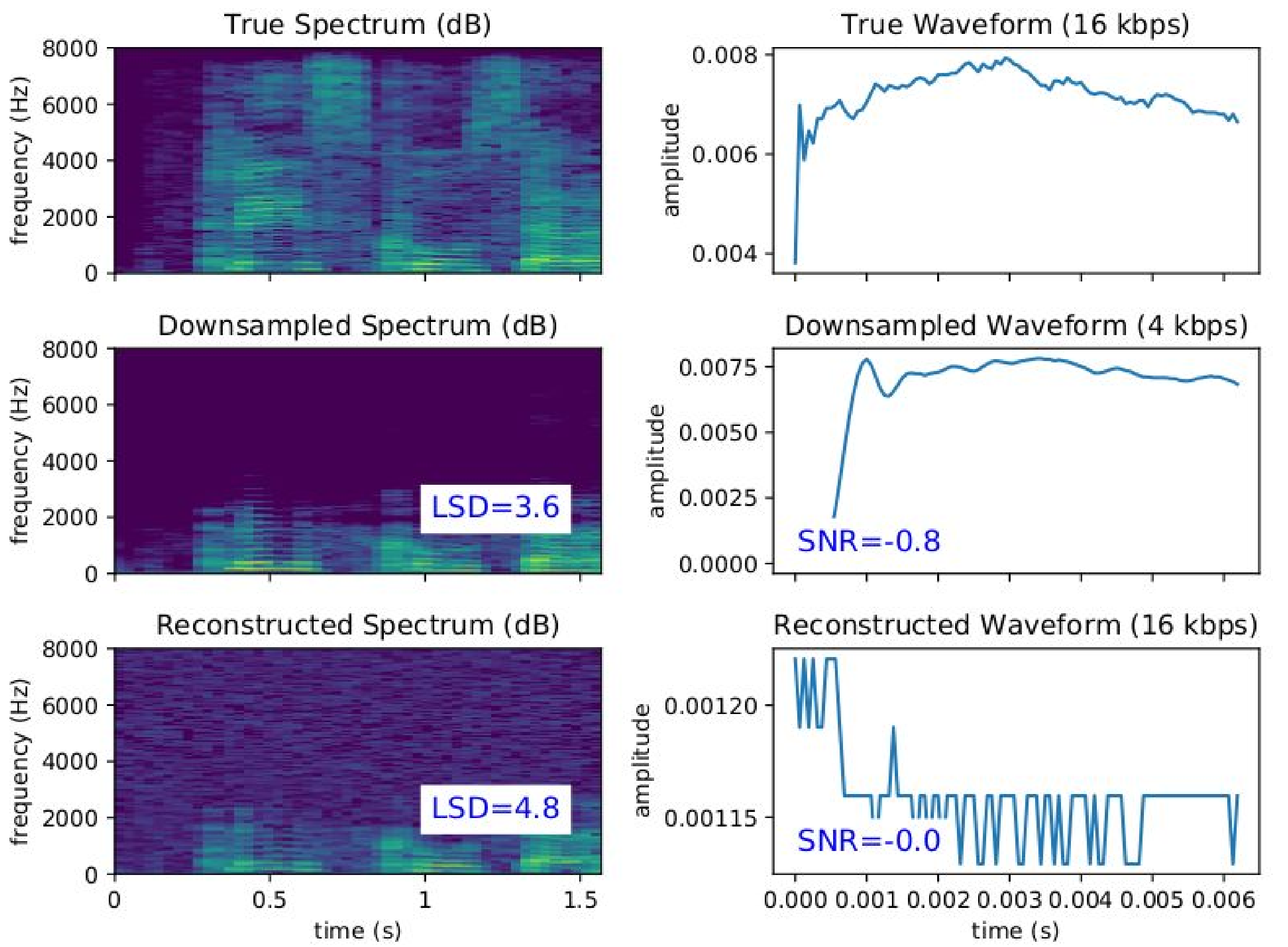
Evaluation Metrics

The performance of the model can be evaluated using the following two features of the input audio sample, actual audio sample and generated super resolved sample

- **Signal to Noise ratio** : Higher the SNR values, the clearer the sound
- **Log Spectral Distance** : Lower the Log Spectral distance indicates matching frequencies

Results


Low Resolution Sample (Input to Neural Net)
High Resolution Sample (True Value)
Output of Neural Network Implementation
Amp
Sample Number/Time
0
0.1
0.2
0.3
0.4
0.5
0.6
5000
5500
6000
6500
7000
7500
8000


True Spectrum (dB)
8000
6000
4000
2000
0
frequency (Hz)
True Waveform (16 kbps)
0.008
0.006
0.004
amplitude
Downsampled Spectrum (dB)
8000
6000
4000
2000
0
frequency (Hz)
LSD=3.6
Downsampled Waveform (4 kbps)
0.0075
0.0050
0.0025
0.0000
amplitude
SNR=-0.8
Reconstructed Spectrum (dB)
8000
6000
4000
2000
0
frequency (Hz)
LSD=4.8
0
0.5
1
1.5
time (s)
Reconstructed Waveform (16 kbps)
0.00120
0.00115
amplitude
SNR=-0.0
0.000
0.001
0.002
0.003
0.004
0.005
0.006
time (s)

Observations and Result Analysis

- ReLU worked fine with respect to audio amplitude, but the results were of lower quality in terms of resolution.
- Sound quality is not that improved (as compared to the HR sample) since a reconstruction based approach was used to fill out missing frequencies and it tried to model the low-res signal itself. Generative models will work better.
- Exploring more loss functions for updation of the weights, and analysing their impact may help to obtain better results.

Future Work

- Implement recent CNN model(2017, Kuleshov et al), try to modify image SRGAN for audio super resolution, Explore LSTM and HMM
- Modify the train-test-validate split to see if results change, explore loss functions
- Use different hyperparameters to analyse the performance.