CS 545 MACHINE LEARNING FOR SIGNAL PROCESSING

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

AUDIO PROCESSING - **RRR**
(RESTORATION, RECOGNITION, REGENERATION)

AUDIO DATA PLAYS A SIGNIFICANT ROLE IN OUR DAILY LIVES, FROM ENTERTAINMENT AND COMMUNICATION TO CRITICAL APPLICATIONS LIKE HEALTHCARE AND SECURITY. THE ABILITY TO CLASSIFY, RESTORE, AND GENERATE AUDIO (MUSIC) SIGNALS HAS A WIDE RANGE OF REAL-WORLD APPLICATIONS, INCLUDING RECOMMENDATION SYSTEMS, AUDIO RESTORATION TOOLS, AND CREATIVE CONTENT GENERATION.

AUTHORS
Pallavi Jain (PJAIN15), Ritik Kulkarni (RK30), Saikrishna Sanniboina (SS235)

INTRODUCTION

The proposed project focuses on implementing advanced signal-processing algorithms for audio data. This project aims to address critical aspects of audio data, including classification, restoration, and generation by leveraging concepts and algorithms such as STFT, Gaussian Classifier, CNN, Wiener filter, Autoregressive model, VAE, GAN, etc.

OBJECTIVE

We aim to achieve a comprehensive understanding of the world of audio signals and the methods to process them for various research and industry use cases.

METHODOLOGY

We started by studying signal processing algorithms that we had learned in class. We decided to try a couple of datasets just to see how each algorithm would perform on different dataset and their different format. The resources we used to implement the algorithms include lecture slides, research papers, and comparison studies.

DATASETS

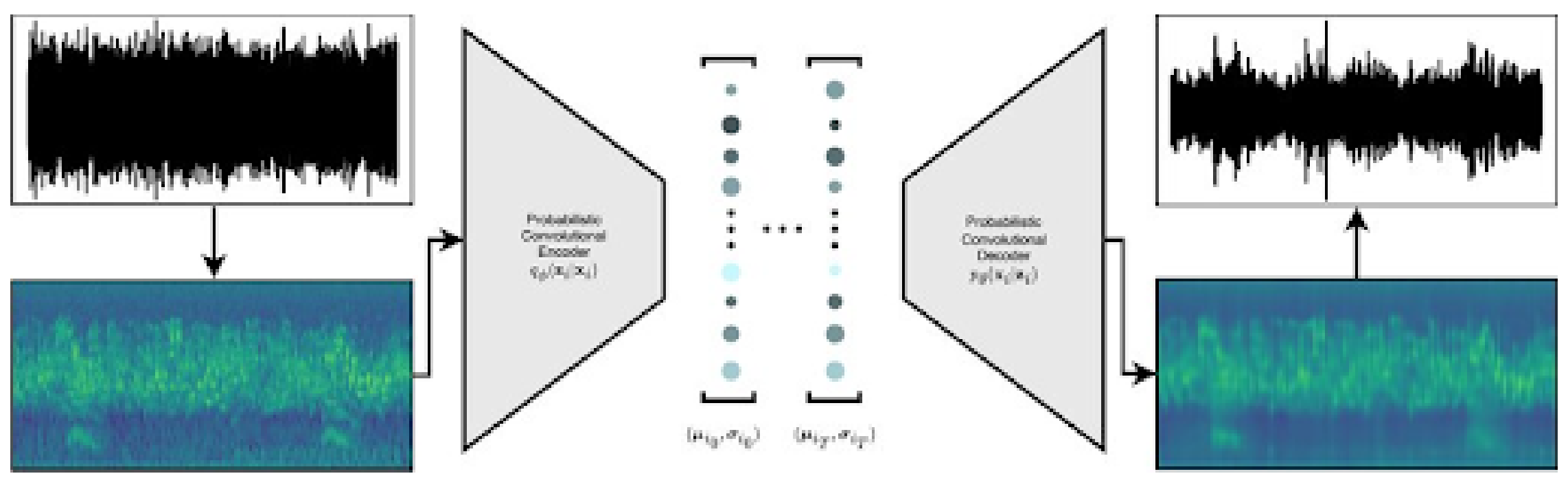
GTZAN Dataset, UrbanSound8K Dataset, Mozilla Common Voice (MCV)

AUDIO GENERATION

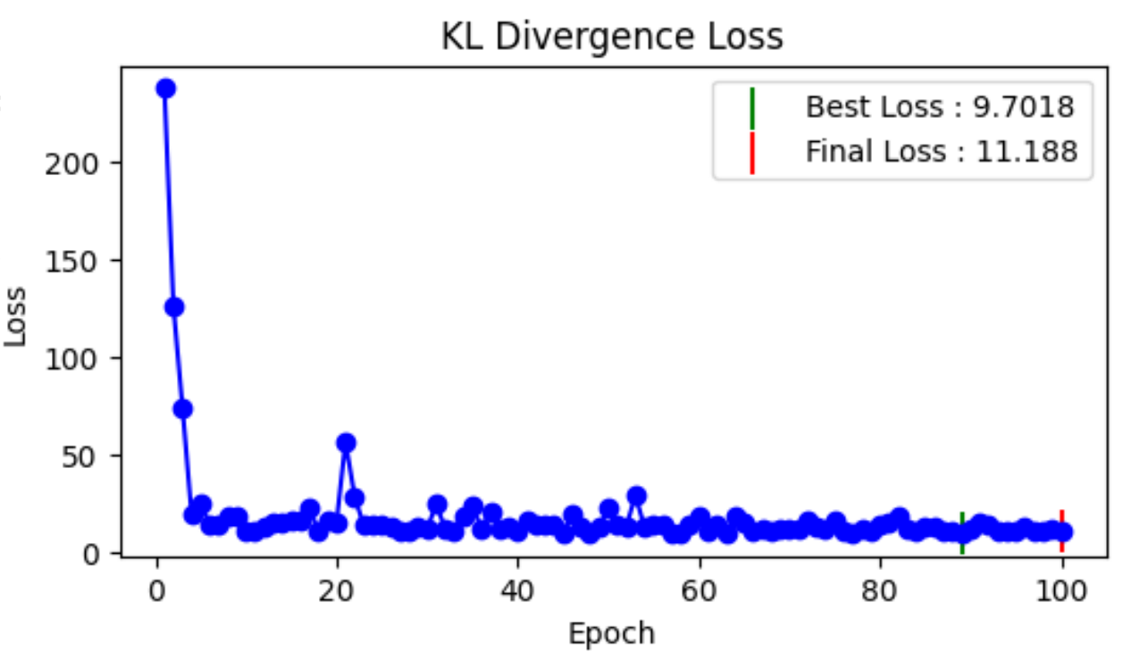
Deep learning is employed for audio generation due to its ability to automatically learn intricate patterns and representations from complex audio data.

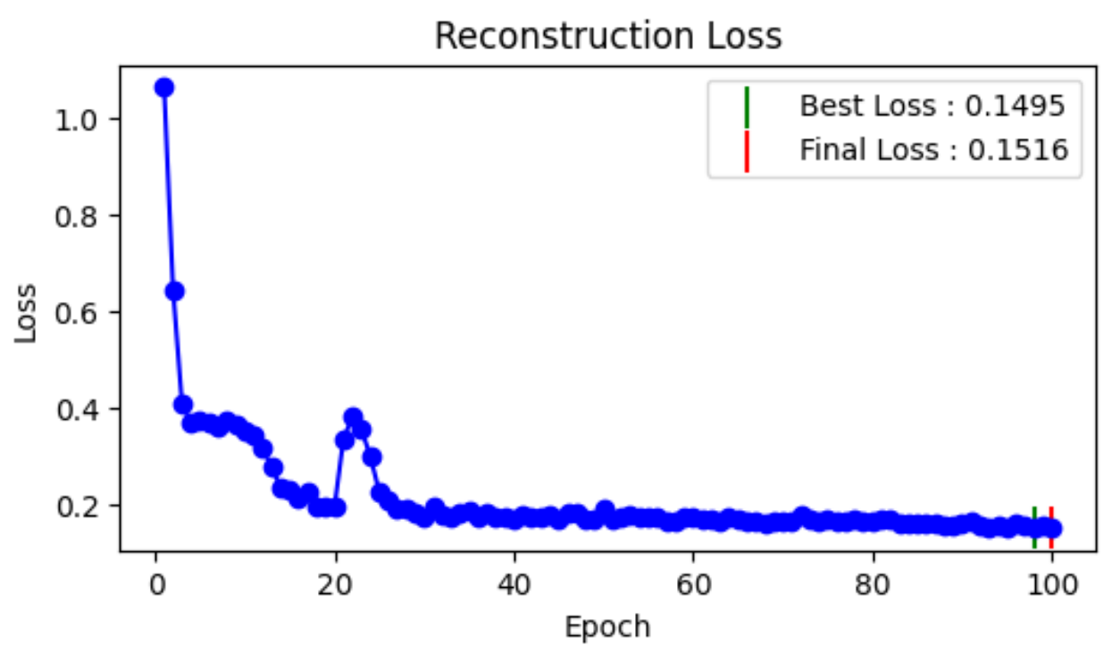
1. VAE (Variational Autoencoder)

We have enhanced the architecture of our Variational Autoencoder (VAE) by incorporating an additional categorical loss for genre prediction. This modification aims to not only generate diverse music but also maintain variety in terms of musical genres. During the inference step, we utilize one-hot encoding for the genre label, providing it as a vector input to the decoder. This enables the model to learn and reproduce specific genre characteristics acquired during training.

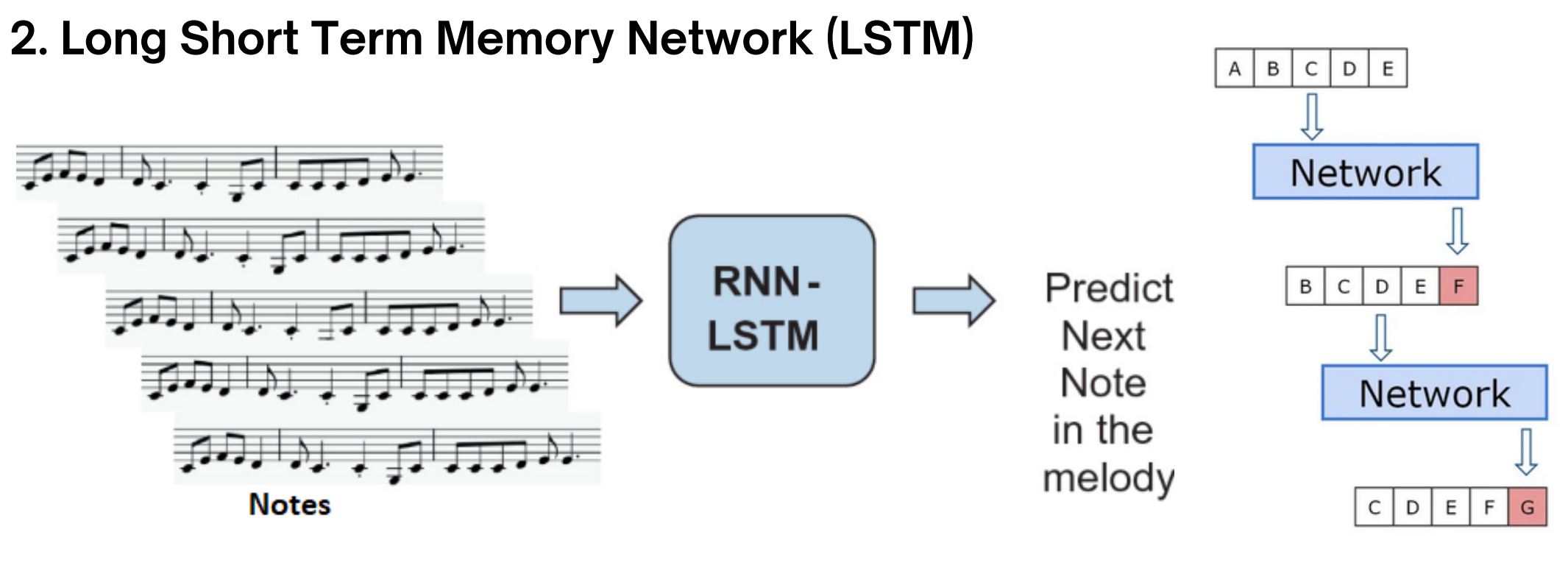

Probabilistic Convolutional Encoder
Probabilistic Convolutional Decoder

Challenges: In our observations, we encountered a couple of challenges. Firstly, the noise introduced became overly dominant, leading to embeddings that lack validity. we found that introducing a weighted loss function (w1*KL + w2*RL) comprising components KL (Kullback-Leibler) divergence and reconstruction loss (RL) resulted in a decrease in the loss values. Still, the generated outputs exhibit characteristics reminiscent of random noise, indicating areas for further refinement.





2. Long Short Term Memory Network (LSTM)



The architecture involves using the Music21 tool for extracting musical notation from MIDI files. For the dataset, we manually curated some Bollywood songs from India which are in MIDI format. MIDI consists of Notes and Chords, where Notes maintain information about pitches, octaves, and offsets and Chords are containers of notes. The training process involves preparing data by loading MIDI files, mapping categorical data to integers, creating input sequences, normalizing input, and one-hot encoding output. The model architecture consists of 3 LSTM layers, 3 dropout layers, 2 dense layers, and an activation layer. Training is performed over 100 epochs.

Epochs	Loss
1	129.77627
100	9.53607

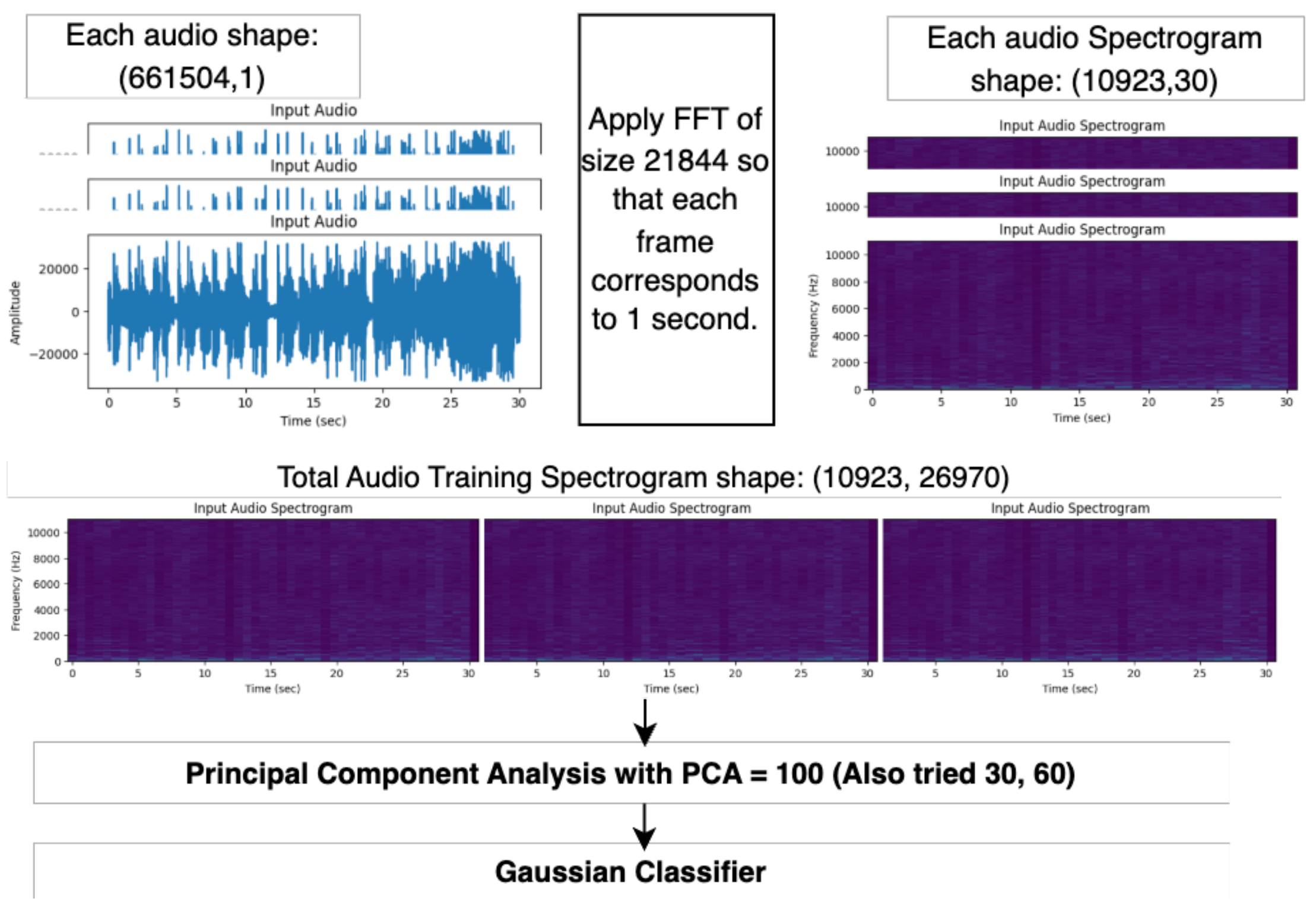
Analysis: After training, the network is used to generate music by predicting the next note in a sequence based on the previous 100 notes. The generated output is then decoded into Note and Chord objects, and a MIDI file is created.

Results show some structure in the generated music, with potential for future improvements, such as supporting varying note durations, adding beginnings and endings to pieces, handling unknown notes, and incorporating more data into the dataset.

AUDIO RECOGNITION

We used the GTZAN music genre dataset for audio recognition tasks. We explored two signal representation techniques and two algorithms to achieve the classification of music pieces into 10 genres, pop, blues, classical, country, disco, hip-hop, jazz, metal, reggae, and rock.

1. Gaussian Classifier models the probability distribution of audio features within different classes using Gaussian distributions. It assumes that the features within each class follow a normal distribution. During recognition, the classifier assigns an input audio sample to the class with the highest likelihood based on the calculated probabilities, making it a probabilistic approach suitable for tasks like music genre identification.


Each audio shape: (661504,1)

Apply FFT of size 21844 so that each frame corresponds to 1 second.

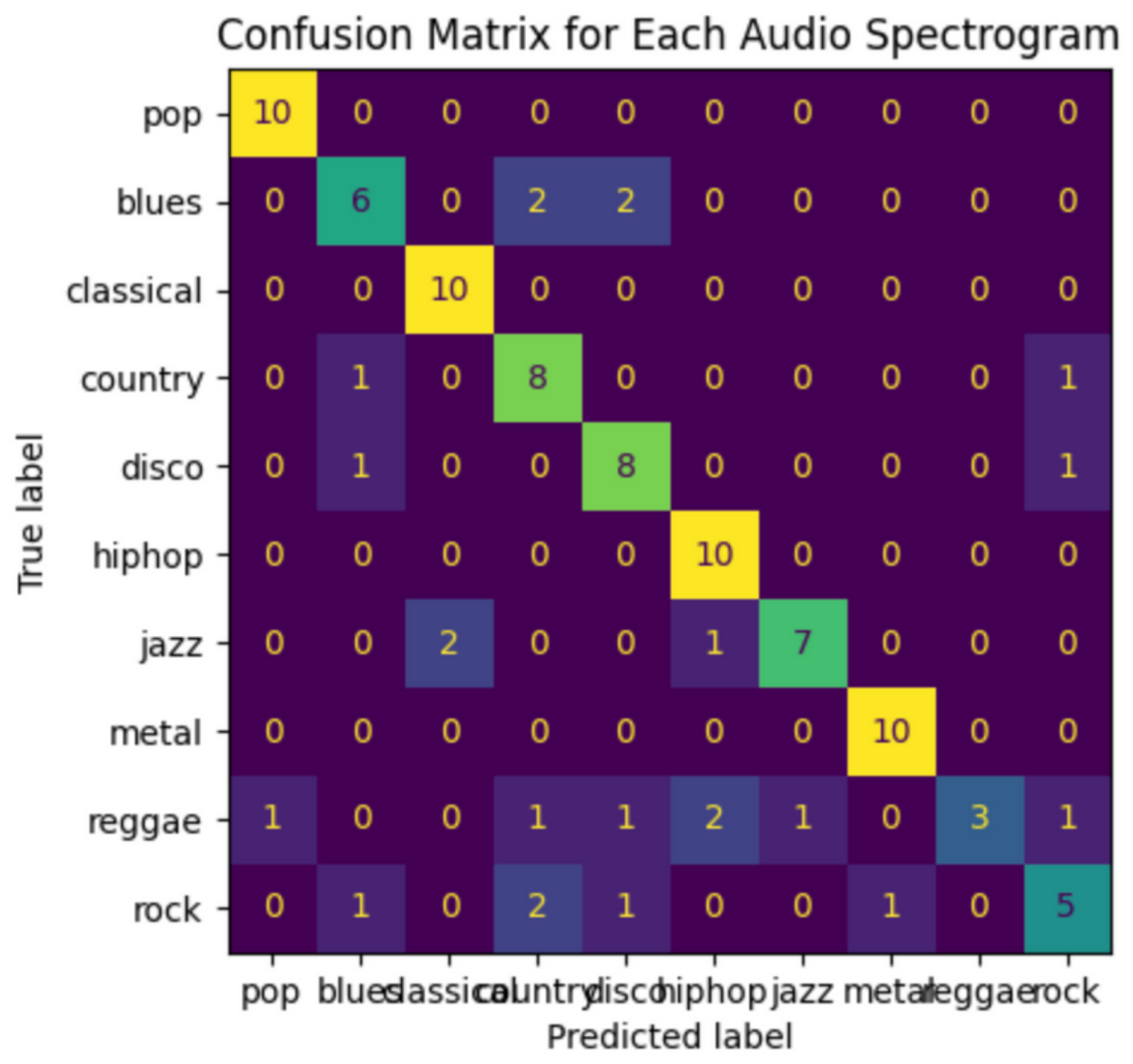
Each audio Spectrogram shape: (10923,30)

Total Audio Training Spectrogram shape: (10923, 26970)

Principal Component Analysis with PCA = 100 (Also tried 30, 60)

Gaussian Classifier

We got the following classification accuracies:
We get 79% accuracy with PCA=90, 75% with PCA=60, 69% with PCA=30.
Analysis: This suggests that a higher dimensional representation (PCA = 100) retains more information and captures more discriminative features, resulting in better classification performance. Also, its most effective application is when there are discernible statistical differences between classes.


Confusion Matrix for Each Audio Spectrogram

True label \ Predicted label	pop	blues	classical	country	disco	hiphop	jazz	metal	reggae	rock
pop	10	0	0	0	0	0	0	0	0	0
blues	0	6	0	2	2	0	0	0	0	0
classical	0	0	10	0	0	0	0	0	0	0
country	0	1	0	8	0	0	0	0	0	1
disco	0	1	0	0	8	0	0	0	0	1
hiphop	0	0	0	0	0	10	0	0	0	0
jazz	0	0	2	0	0	1	7	0	0	0
metal	0	0	0	0	0	0	0	10	0	0
reggae	1	0	0	1	1	2	1	0	3	1
rock	0	1	0	2	1	0	0	1	0	5

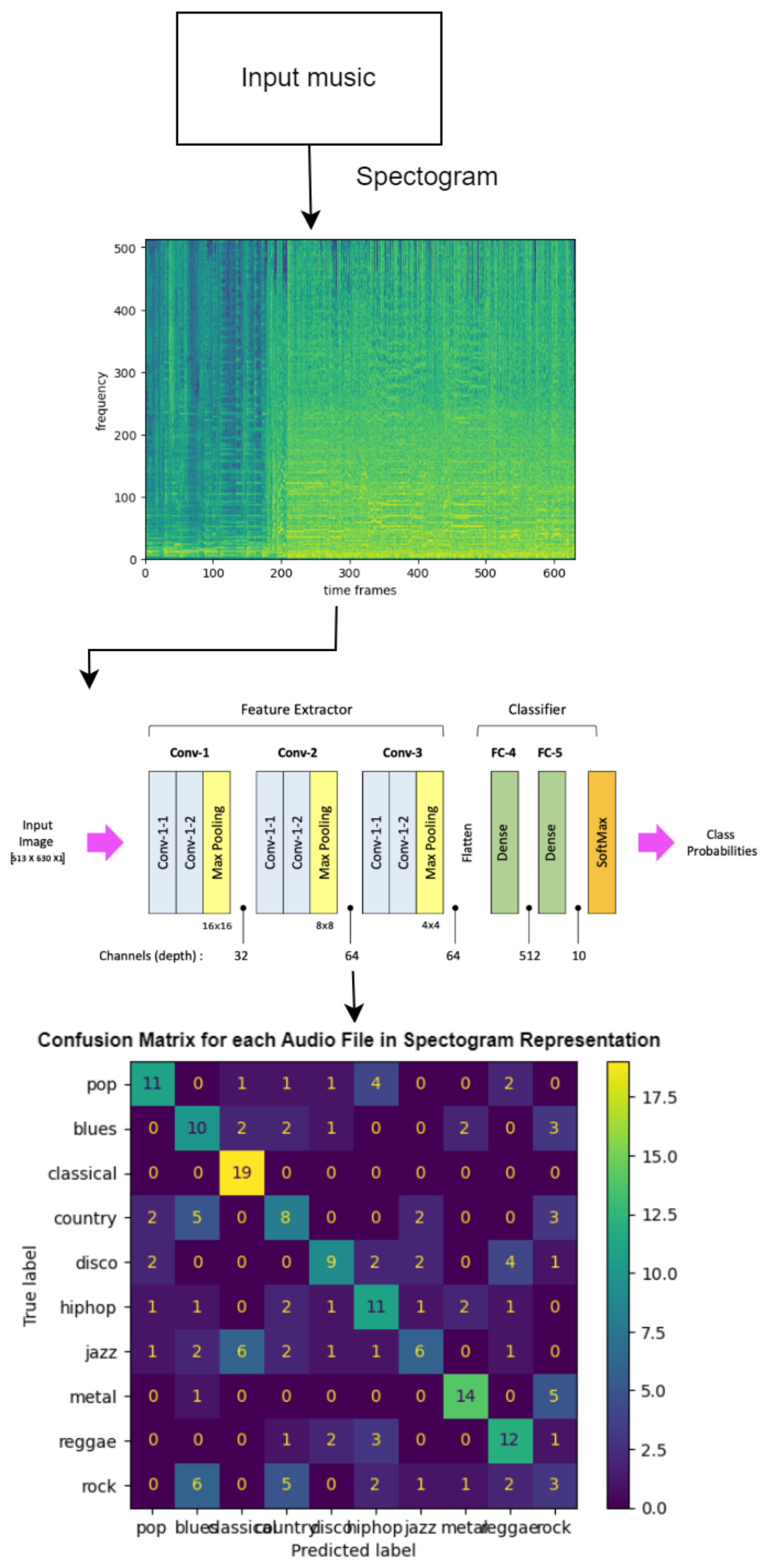
2. CNN Leveraging CNNs, renowned for image processing, our architecture treats audio spectrograms (513 x 630 x 1) as images. This approach exploits the visual pattern recognition strength of CNNs for effective audio analysis. The 513 x 630 x 1 format encapsulates the frequency and time dimensions, enhancing feature extraction.

Analysis:
Using basic CNN model with just spectrogram features, measured 69% accuracy on the test data set of 330 files. CNN can effectively denoise speech with a smaller network size according to its weight-sharing property.

Extracted More Features from audio and concatenated them for a more enriched representation of audio.

1. MFCC
   - Enhances audio signal representation with coefficients capturing short-term power spectrum.
2. Zero Crossing Rate:
   - Measures the rate of sign changes, indicating audio noisiness or percussiveness.
3. Spectral Centroid:
   - Represents the "centre of mass" of the spectrum, offering insights into audio characteristics.
   -

Measured 65% accuracy on the test data set of 330 files. This extra information is further deviating model from its learning Path.

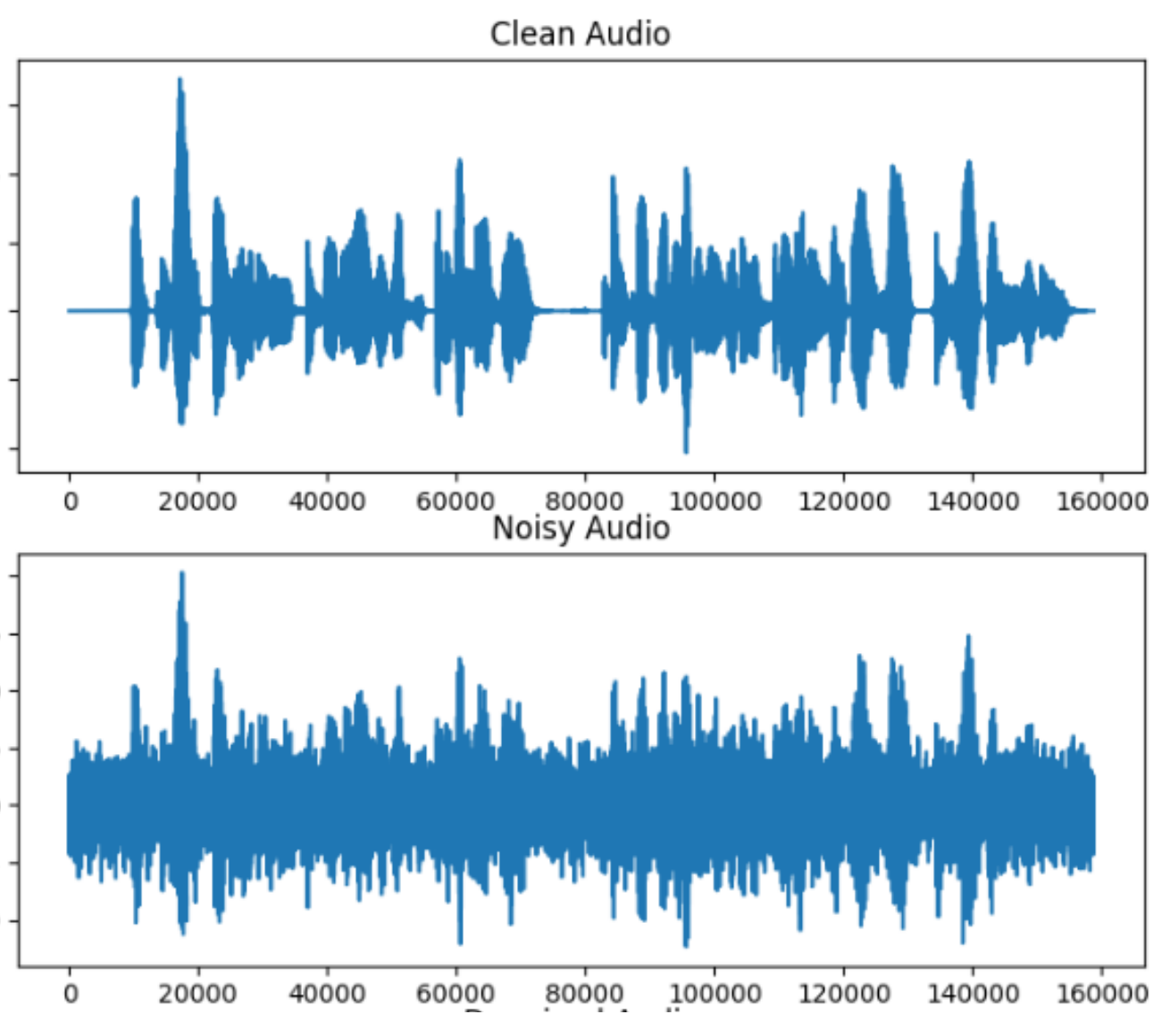

Input music

Spectogram

Confusion Matrix for each Audio File in Spectogram Representation

True label \ Predicted label	pop	blues	classical	country	disco	hiphop	jazz	metal	reggae	rock
pop	11	0	1	1	1	4	0	0	2	0
blues	0	10	2	2	1	0	0	2	0	3
classical	0	0	19	0	0	0	0	0	0	0
country	2	5	0	8	0	0	2	0	0	3
disco	2	0	0	0	9	2	2	0	4	1
hiphop	1	1	0	2	1	11	1	2	1	0
jazz	1	2	6	2	1	1	6	0	1	0
metal	0	1	0	0	0	0	0	14	0	5
reggae	0	0	0	1	2	3	0	0	12	1
rock	0	6	0	5	0	2	1	1	2	3

AUDIO RESTORATION

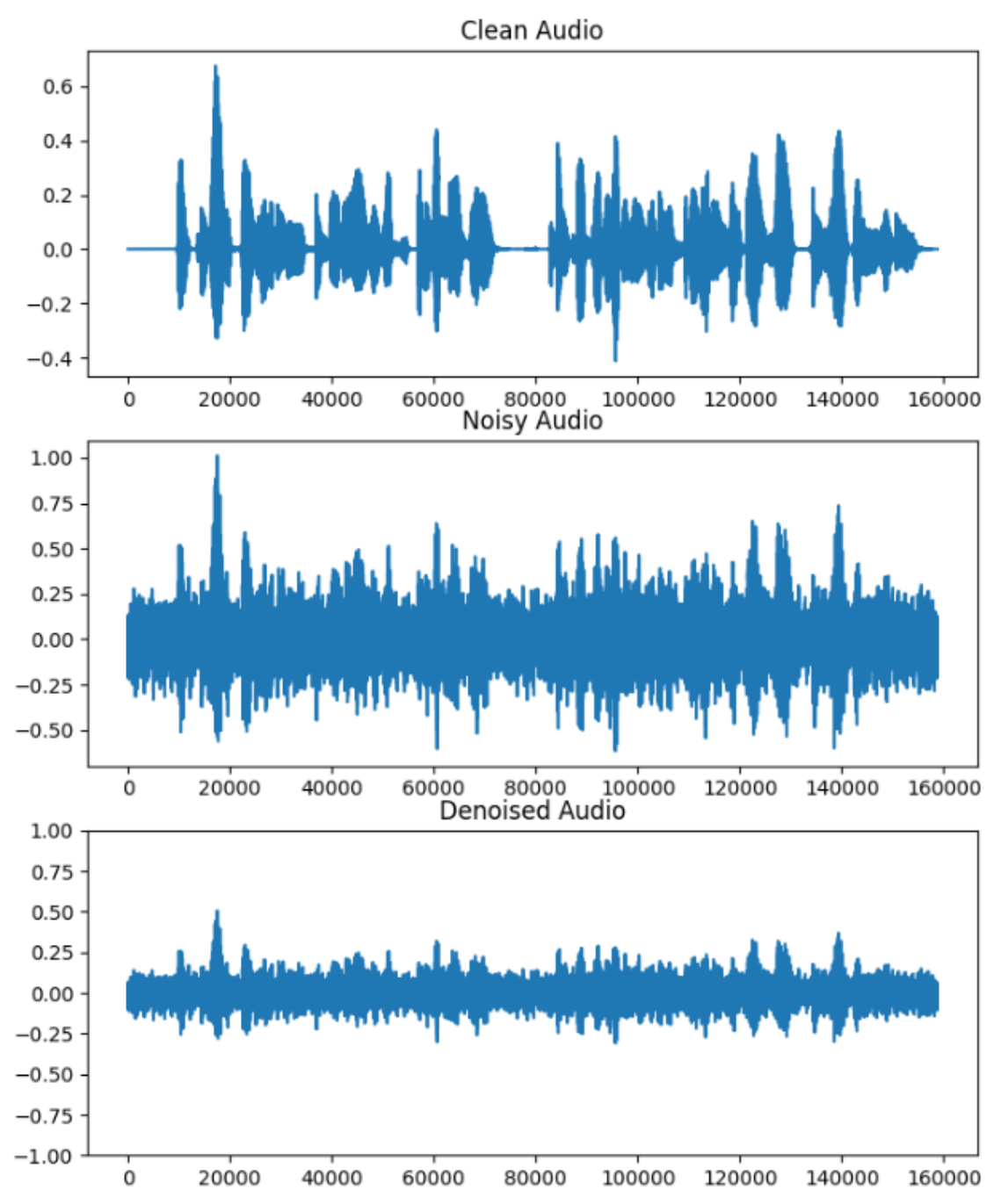


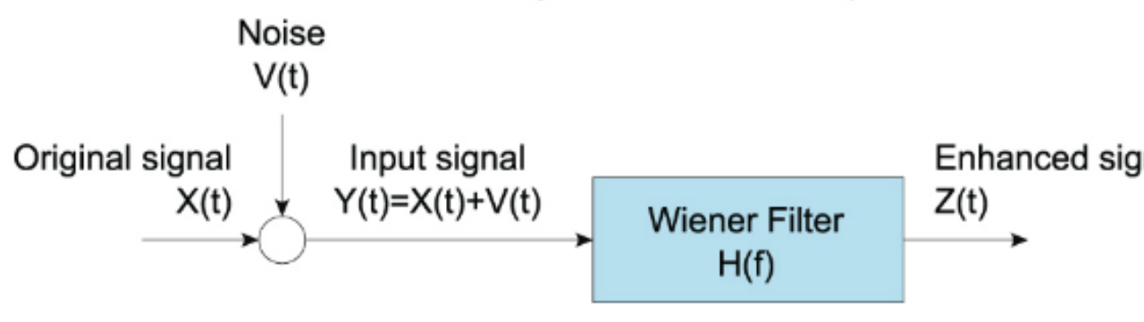
We used the MCV and UrbanSound8K datasets for achieving a **denoising** task. We first take a 6-second audio from MCV and introduce a 4-second noise into it from the UrbanSound8K dataset.

We implement a simple Wiener filter as the first technique to explore and understand the process of denoising audio signals. Next, we implement an Auto-regressive model from a research paper.

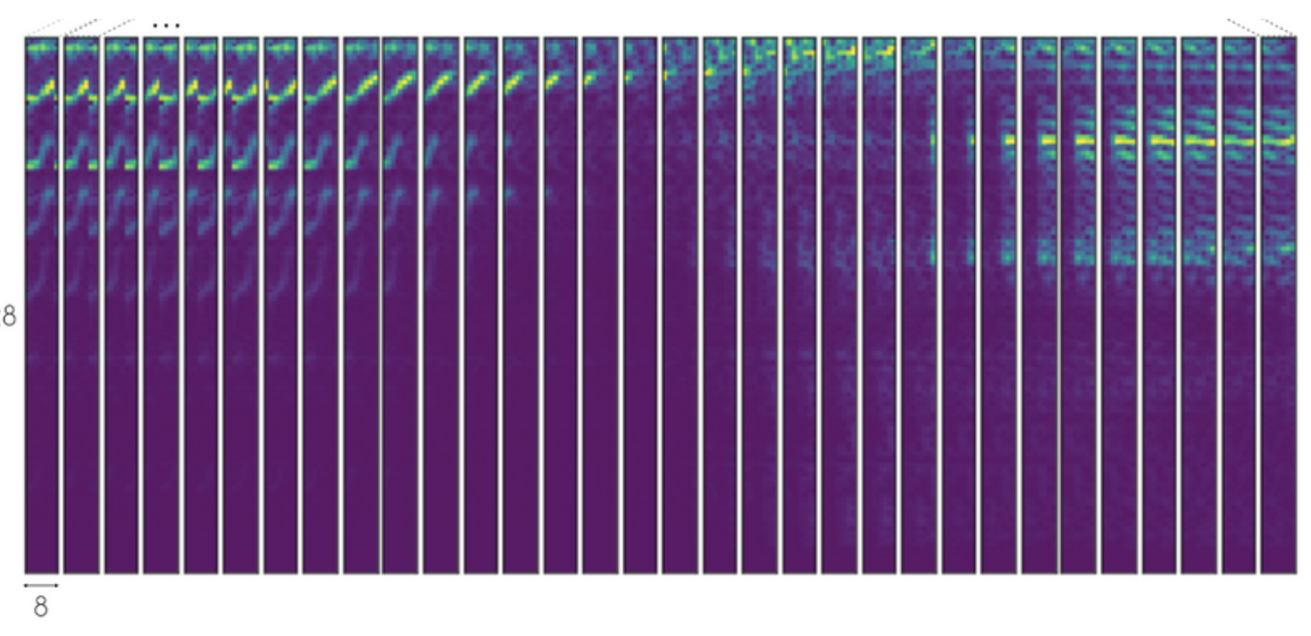
1. Wiener Filter's primary goal is to enhance or restore a signal that has been corrupted by noise. It works by estimating the original signal from the noisy observation, taking into account the statistical properties of both the signal and the noise. The filter is designed to minimize the mean square error between the noise-free signal and the original signal.

Analysis: The wiener filter aims to emphasize the frequencies where the signal is strong and suppress the frequencies where the noise dominates. This leads to the wiener filter reducing noise while compromising some of the signal's integrity. Since it suppresses the frequencies where noise dominates, leading to some signal degradation, the more the noise, the more the audio signal gets degraded. MSE = 0.0023

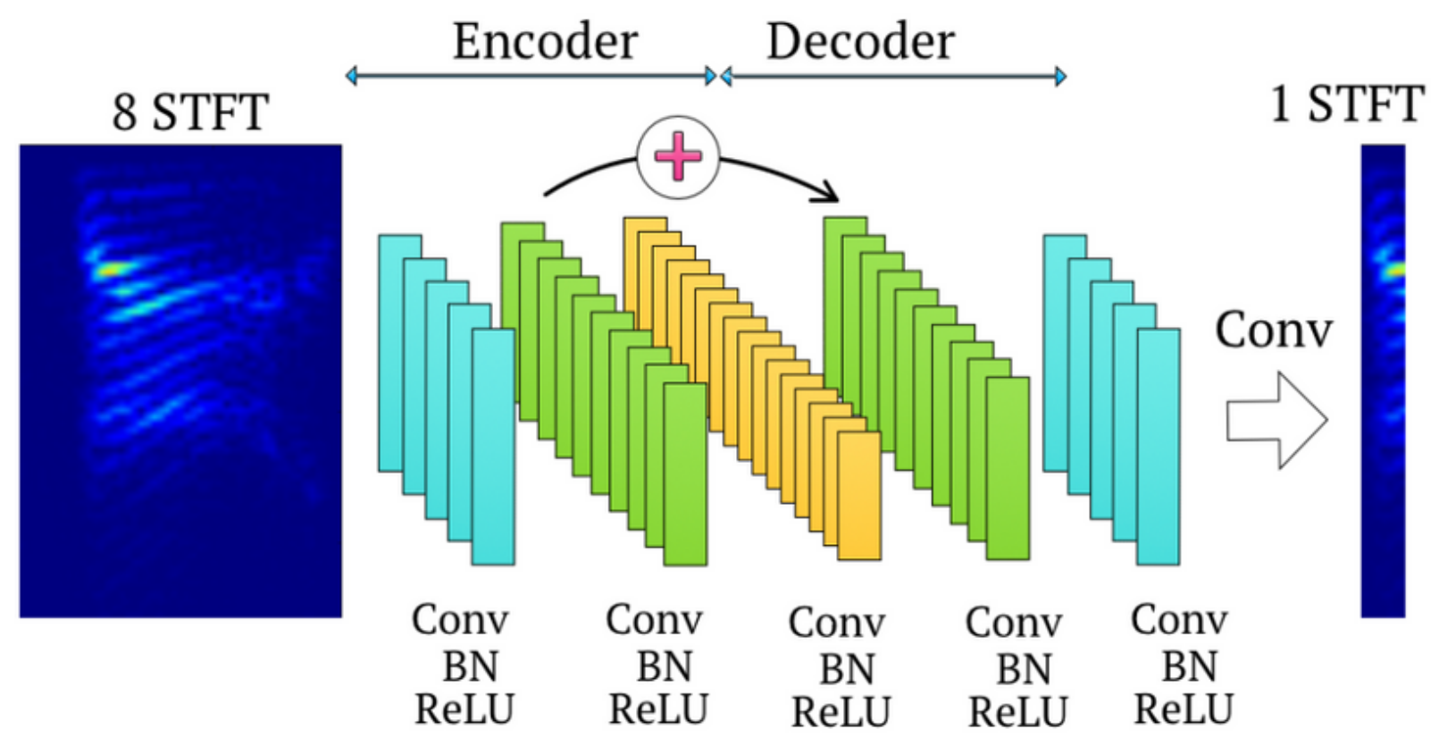


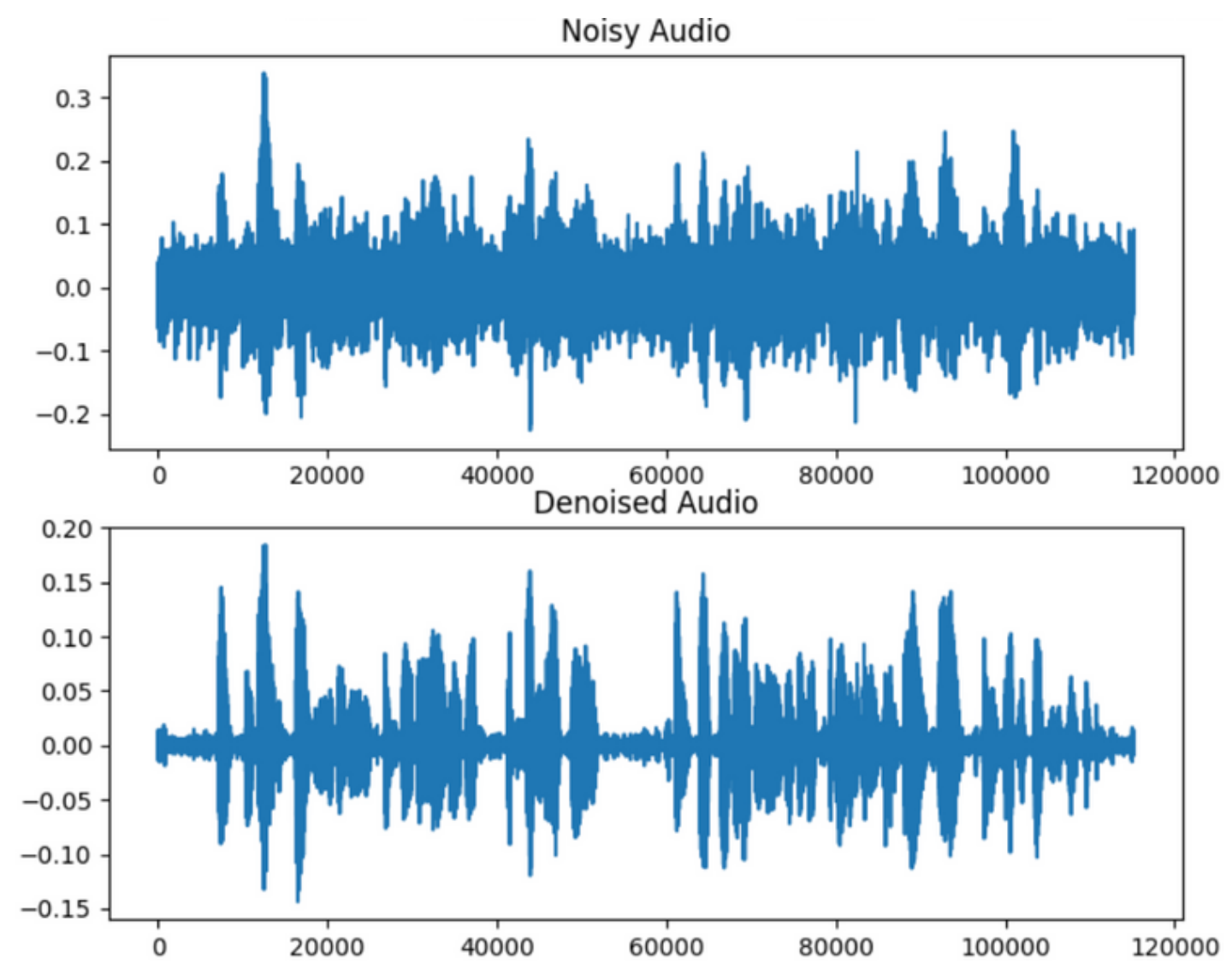


2. Auto-Regressive Model from the paper- "A fully convolutional neural network for speech enhancement". Here, the authors proposed a Cascaded Redundant Convolutional Encoder-Decoder Network (CR-CED). The model is based on symmetric encoder-decoder architectures. Both components contain repeated blocks of Convolution, ReLU and Batch Normalization.



We use a periodic hamming window of length 256 and a hop size of 64 for the STFT window, ensuring a 75% overlap. Input vectors are formed by concatenating 8 consecutive noisy STFT vectors, resulting in a shape of (129, 8). The model is an autoregressive system, that predicts the current signal based on past observations. Targets consist of a single STFT frequency representation with a shape of (129,1) derived from the clean audio.





Analysis:
We observed that the success of R-CED is associated with the increasing dimension of the feature space along encoder and decreasing dimension along the decoder. In most of the examples, the model manages to smooth the noise but it doesn't get rid of it completely. Another important point is that audio signals are non-stationary. In other words, the signal's mean and variance are not constant over time. Thus, there is not much sense in computing a Fourier Transform over the entire audio signal.

CONCLUSION

In this comprehensive exploration of audio processing, we delved into three crucial facets: recognition, restoration (denoising), and regeneration (music generation). Through rigorous implementation, we witnessed the efficacy of various approaches within each domain. Recognition models exhibited the potential to accurately identify and categorize audio content. Denoising techniques showcased their ability to effectively remove unwanted noise, enhancing audio clarity. Meanwhile, in the realm of music generation, innovative methods demonstrated the capacity to recreate and regenerate harmonious compositions. Collectively, these findings underscore the versatility and transformative power of audio processing, offering promising avenues for real-world applications and further advancements in the field.

References: https://sthalles.github.io/practical-deep-learning-audio-denoising/ , https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification, https://ieeexplore.ieee.org/document/1643650