Neural Audio Compressor — Extended Project Description and Implementation Plan

PROJECT OVERVIEW

The Neural Audio Compressor is a hybrid Digital Signal Processing (DSP) and Deep Learning system designed to compress audio signals into a compact latent representation and reconstruct them with high fidelity. The goal is to build a modern neural audio codec similar in spirit to systems like SoundStream or EnCodec, but in a simplified research-friendly form. The project demonstrates mastery of audio feature engineering, spectrum analysis, neural network design, and evaluation of perceptual signal quality.

The workflow:

1) Raw audio waveform is transformed into a time-frequency representation (STFT or log-Mel).

2) A neural encoder compresses the spectrogram into a latent vector.

3) A decoder reconstructs the spectrogram from the latent space.

4) ISTFT (or Mel inversion) reconstructs the waveform.

5) Multiple metrics evaluate reconstruction quality, both objective and perceptual.

APPLICATIONS

- Speech codecs and audio compression

- On-device low■bitrate audio transmission

- Feature learning for audio generation models

- Music and instrument reconstruction

- Research in representation learning and signal processing

---

1. CLEAN PROJECT STRUCTURE (ASCII-ONLY BULLETS)

Project folders:

- data/

- configs/

- src/

- datasets/

- transforms/

- models/

- training/

- evaluation/

- inference/

- experiments/

- notebooks/

- results/

- README.md

This structure avoids Unicode characters that could appear as black boxes in PDF.

---

2. DATA PREPARATION

Datasets:

- VCTK or LibriSpeech for speech

- NSynth for musical instruments

- Custom WAV files for mixed data

Preprocessing:

- Convert to mono

- Resample to 16000 Hz or 22050 Hz

- Normalize amplitude to [-1, 1]

- Cut or pad audio to fixed length (1–3 seconds)

The AudioDataset class loads audio, applies transforms, generates STFT or log-Mel, and prepares tensors for training.

---

3. SIGNAL PROCESSING PIPELINE

Short-Time Fourier Transform (STFT):

- n_fft = 1024
- hop_length = 256
- win_length = 1024
- Hann window

Outputs:

- Magnitude |S|
- Phase angle(S)

Log-Domain:

- log_mag = log(1 + |S|)

Alternative Representations:

- Mel spectrogram
- Log-Mel
- MFCC

These representations are fundamental DSP tools and form the basis for the neural model.

---

4. AUTOENCODER ARCHITECTURE

ENCODER:

- Conv2D layers with stride=2 for downsampling
- Increasing channel depth: 1→32→64→128
- Flatten + Linear → latent_vector

DECODER:

- Linear → reshape

- ConvTranspose2D to upsample step-by-step

- Final output: reconstructed magnitude spectrogram

Compression:

- latent_dim options: 64 / 128 / 256 / 512

- Compression ratio = (input_size / latent_dim)

---

5. LOSS FUNCTIONS

1. L1 Reconstruction Loss

2. Multi-Scale STFT Loss

3. Mel-Spectrogram Loss (perceptual)

4. Combined loss:

Loss = a * L1 + b * STFTLoss + c * MelLoss

The multi-scale losses help stabilize frequency■domain reconstruction and improve subjective audio quality.

---

6. TRAINING

Configuration:

- Batch size: 8–32

- Epochs: 50–200

- Optimizer: AdamW

- Learning rate: 1e-3 with scheduler

Logging:

- Loss curves

- Reconstructed samples

- Spectrogram comparisons

- Model checkpoints

---

7. EVALUATION

Objective metrics:

- SNR (Signal-to-Noise Ratio)

- SDR (Signal-to-Distortion Ratio)

- Log■spectral distance

- MSE / L1 in waveform space

Visual evaluation:

- Spectrogram comparisons

- Error heatmaps

Subjective evaluation:

- Listening tests for artifacts such as:

- metallic ringing

- smearing

- aliasing

- muffled frequencies

---

8. EXPERIMENTS

1. Latent dimension comparison

2. Representation comparison (STFT vs log-STFT vs log-Mel)

3. Loss function comparison

4. Architecture comparison:

- Conv AutoEncoder
- U-Net AE
- Transformer AE
- Variational AutoEncoder

---

9. FINAL DOCUMENTATION AND DEMO

README includes:

- Full architecture explanation
- DSP background
- Training instructions
- Audio demo samples
- Spectrogram visuals

Demo notebook:

- Load pretrained model
- Encode/decode audio sample
- Visualize reconstruction

---

10. FUTURE IMPROVEMENTS

- Switch to VAE for probabilistic latent space
- Add vector quantization (VQ-VAE style)
- Predict complex spectrum instead of magnitude only
- Introduce GAN loss for perceptual realism
- Explore end-to-end waveform models (WaveNet, Diffusion)

END OF DOCUMENT