Echoes Reborn: A Comprehensive Pipeline for Historic Piano Audio Restoration

Johanna Smriti
Department of Computer Science and Engineering
University of California, Santa Cruz
ijegan@ucsc.edu

Abstract

Historical piano recordings hold immense cultural value, but often suffer from degradation, including gramophone noise, gaps in audio, and loss of high-frequency details. This paper introduces an end-to-end neural pipeline that integrates classification, de-noising, and audio inpainting to restore these recordings. Our approach uniquely addresses the challenge of preserving valuable signals and melodies even when the input audio lacks noise, which is a common limitation in traditional de-noising models. This ensures high fidelity and integrity in the restored audio output. We demonstrate our method's efficacy using MusicNet's historic piano dataset, attaining a test accuracy of 99.92% for classification and a Signal-to-Noise Ratio (SNR) improvement of 21.21 dB in de-noising tasks.

1 Introduction

Restoring historical audio is essential for preserving the cultural and musical heritage of early recordings, which are often marred by degradation such as gramophone noise, loss of high frequencies, and audio gaps. As Michael De Sapio [1] observes, preserving classical music is key to connecting with the emotions and ideals of the past, highlighting the significance of restoring historical performances. To address these challenges, we introduce Echoes Reborn, a unified, end-to-end neural pipeline designed specifically to restore and enhance historic piano recordings. Our approach prioritizes both the integrity of the original sound and the preservation of musical details. Notably, our pipeline incorporates a novel audio processing method that ensures noise-free audio remains untouched during denoising, overcoming a common limitation of existing methods that can degrade the quality of pristine recordings through overprocessing.

2 Related Work

The problem of audio restoration has been the subject of extensive study. Early pioneering work by Takayuki Sasaki in the 1980s [2] demonstrated how temporal localization of noise and auditory rearrangement techniques could help reduce distortion in speech and music recordings. Sasaki's experiments laid the groundwork for modern restoration methods by emphasizing the importance of context-sensitive noise removal.

In recent years, advances in machine learning have led to more sophisticated methods for restoring audio. Notably, methods such as those developed by E. Moliner and V. Välimäki [3, 4] have incorporated deep learning models, like convolutional neural networks (CNNs), to perform end-to-end restoration tasks, achieving substantial improvements in de-noising and enhancement. For instance, E. Moliner [3] introduced a diffusion-based generative equalizer (BABE-2), which enhances historical music recordings by utilizing priors from diffusion models, while Moliner & Välimäki [4]

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

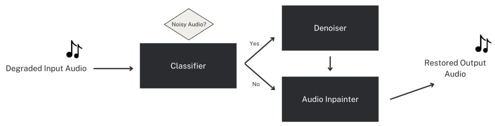

Figure 1: End-to-End Architecture

presented a two-stage U-Net model specifically designed for high-fidelity de-noising of historical recordings.

3 Methodology

Our proposed method integrates three components: a classifier, de-noiser, and an audio inpainter into a unified pipeline to effectively restore degraded historic piano audio, as shown in Figure 1.

3.1 Classifier Design

The classifier determines whether an audio sample is noisy or clean, enabling conditional de-noising only when necessary.

3.1.1 Architecture

The classifier is designed to distinguish between clean and noisy audio samples. It uses a convolutional neural network (CNN) with the following structure, as shown in Figure 2:

Convolutional Layers: The model includes three convolutional layers, defined as:

$$\mathbf{h}_l = \sigma(\mathbf{W}_l * \mathbf{x}_l + \mathbf{b}_l), \quad l \in \{1, 2, 3\}$$

where:

- $\mathbf{h}_l$ is the output feature map at layer l.
- $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and biases of the l-th convolutional layer.
- $*$ denotes the convolution operation.
- σ is the ReLU activation function, defined as $\sigma(x) = \max(0, x)$.

The layers are configured as follows:

- **Conv1**: 1 input channel (for grayscale spectrograms), 16 filters, kernel size of 3×3, stride 1, and padding 1.
- **Conv2**: Takes the 16-channel output from the first layer, with 32 filters and the same kernel configuration.
- **Conv3**: Processes the 32-channel output, producing 64 filters.

Pooling and Activation: Each convolutional layer is followed by a max-pooling operation:

$$\mathbf{h}'_l = \text{MaxPool}(\mathbf{h}_l, k = 2, s = 2)$$

where k is the kernel size and s is the stride. This operation halves the spatial dimensions and reduces computational complexity.

Dropout: To prevent overfitting, a dropout layer is applied with a rate of $p = 0.3$:

$$\mathbf{h} = \text{Dropout}(\mathbf{h}, p)$$

where p represents the probability of dropping a unit.

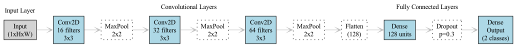

Figure 2: Classifier Architecture

Fully Connected Layers: After the convolutional layers, the output is flattened into a 1D vector:

$$\mathbf{x}_{\text{flat}} = \text{Flatten}(\mathbf{h}'_3)$$

This vector is processed through two fully connected layers:

- **Dynamic Feature Mapping:** A linear transformation maps the flattened vector to 128 dimensions:
$$\mathbf{z}_1 = \sigma(\mathbf{W}_1 \mathbf{x}_{\text{flat}} + \mathbf{b}_1)$$
- **Output Layer:** Another linear transformation maps the 128-dimensional features to two classes:
$$\mathbf{z}_2 = \mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2$$

The final output $\mathbf{z}_2$ is passed through a softmax function for classification:

$$\hat{y} = \text{Softmax}(\mathbf{z}_2)$$

Forward Pass: The input spectrogram is processed sequentially through convolutional layers, pooling, and activation functions:

$$\hat{y} = f_{\text{CNN}}(\mathbf{x})$$

where f_{CNN} represents the entire CNN pipeline.

3.1.2 Performance Evaluation

The classifier's performance was evaluated using metrics such as accuracy:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

and average loss during testing. Testing confirmed the classifier's reliability in distinguishing between clean and noisy audio, achieving a test accuracy of 99.92%.

3.2 De-noising Model Design

The de-noising model restores clean audio signals from noisy spectrogram inputs, leveraging a Convolutional Autoencoder (CAE) architecture. It employs feature extraction in the encoder, followed by reconstruction in the decoder. The model is optimized to minimize loss metrics while enhancing perceptual quality.

3.2.1 Architecture

The autoencoder consists of an encoder-decoder structure, as shown in Figure 3.

Encoder: The encoder reduces the dimensionality of input features while preserving meaningful representations:

$$\mathbf{h}_l = \sigma(\mathbf{W}_l * \mathbf{x}_l + \mathbf{b}_l), \quad l \in \{1, 2\}$$

where:

- $\mathbf{x}_l$ is the input at layer l.
- $\mathbf{W}_l$ and $\mathbf{b}_l$ are weights and biases for layer l.
- $*$ denotes convolution.
- σ is the ReLU activation function, $\sigma(x) = \max(0, x)$.

Max-pooling is applied after the convolutional layers:

$$\mathbf{h}'_l = \text{MaxPool}(\mathbf{h}_l, k = 2, s = 2)$$

where k is the kernel size and s is the stride.

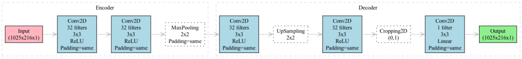
Encoder
Decoder
Input (1025x216x1)
Conv2D 32 filters 3x3 ReLU Padding=same
Conv2D 32 filters 3x3 ReLU Padding=same
MaxPooling 2x2 Padding=same
Conv2D 32 filters 3x3 ReLU Padding=same
UpSampling 2x2
Conv2D 32 filters 3x3 ReLU Padding=same
Cropping2D (0,1)
Conv2D 1 filter 3x3 Linear Padding=same
Output (1025x216x1)

Figure 3: Audio De-noiser Architecture

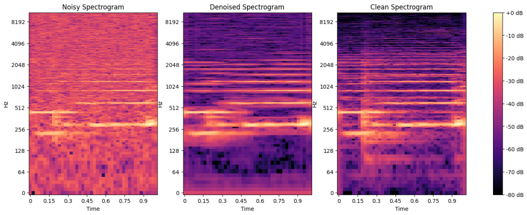

Figure 4: Visualization and Results

Decoder: The decoder reconstructs the original spectrogram from the compressed latent representation:

$$\mathbf{y}_l = \sigma(\mathbf{W}'_l * \mathbf{h}_l + \mathbf{b}'_l), \quad l \in \{3, 4\}$$

Upsampling layers increase spatial dimensions:

$$\mathbf{y}'_l = \text{UpSample}(\mathbf{y}_l, k = 2)$$

Cropping is applied to align the dimensions of reconstructed spectrograms:

$$\mathbf{y}_{\text{crop}} = \text{Crop}(\mathbf{y}'_l, \text{padding})$$

The final output is:

$$\hat{\mathbf{x}} = \sigma(\mathbf{W}_5 * \mathbf{y}_{\text{crop}} + \mathbf{b}_5)$$

Model Parameters: The autoencoder has 28,353 parameters:

- **Input Shape:** $(1025, 216, 1)$
- **Latent Features:** Encoded into $(513, 108, 32)$
- **Output Shape:** $(1025, 216, 1)$

3.2.2 Performance Evaluation

The model's performance was evaluated on the test set using multiple metrics:

- **Validation Loss:** Improved to 0.0491 after 50 epochs.
- **MAE:** Achieved a mean absolute error of 0.0834.
- **SNR:** Recorded an average signal-to-noise ratio of 21.21 dB.
- **STFT Loss:** Calculated based on spectrogram differences:

$$\mathcal{L}_{\text{STFT}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{S}(\mathbf{x}_i) - \mathcal{S}(\hat{\mathbf{x}}_i)\|_1$$

where $\mathcal{S}(\cdot)$ represents the Short-Time Fourier Transform (STFT).

3.2.3 Visualization and Results

Figure 4 visualizes the noisy, clean, and de-noised spectrograms. The model's outputs closely match the clean spectrograms, demonstrating its efficacy in audio restoration.

The model successfully restores high-fidelity audio signals, as confirmed by both numerical metrics and perceptual evaluations.

3.3 Audio Inpainting

Audio inpainting aims to reconstruct missing segments in an audio signal, preserving continuity and quality. This section details an approach using Linear Predictive Coding (LPC) to inpaint silent gaps. The mathematical method utilized in this work aligns with the approach described by Andrés Marafioti [5], where deep neural networks (DNNs) were employed for audio inpainting tasks using time-frequency (TF) coefficients to address gaps in music signals.

3.3.1 Linear Predictive Coding (LPC)

Linear Predictive Coding (LPC) predicts future samples of a signal based on its past values. It is a widely-used method for modeling audio signals, effectively capturing their spectral envelope through prediction coefficients.

Key Components:

- **Prediction Coefficients:** Represent the signal properties and are computed to minimize the prediction error.
- **Error Signal:** The difference between the predicted and actual signal, minimized during LPC analysis.

The signal at sample n, $x[n]$, is expressed as:

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + e[n]$$

where:

- a_k: Prediction coefficients.
- $e[n]$: Prediction error.
- p: Order of the LPC model.

3.3.2 Audio Inpainting Using LPC

LPC-based audio inpainting proceeds as follows:

1. Silent Gap Detection: Identify silent gaps using a sliding window to calculate the mean amplitude:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} |x[i]|$$

Regions with $\mu <$ threshold are marked as gaps.

2. Context Extraction: Extract audio segments before and after the gap. Compute LPC coefficients using Burg's algorithm to predict forward and backward signals:

$$x_{\text{forw}}[n] = -\sum_{k=1}^{p} a_k x[n-k], \quad x_{\text{backw}}[n] = -\sum_{k=1}^{p} b_k x[n-k]$$

3. Signal Reconstruction: Blend the forward and backward predictions using a cosine-weighted function:

$$x_{\text{blend}}[n] = \cos^2\left(\frac{\pi n}{2L}\right) x_{\text{forw}}[n] + \cos^2\left(\frac{\pi(L-n)}{2L}\right) x_{\text{backw}}[n]$$

where L is the gap length. Combine the reconstructed gap with the original audio to produce the final output.

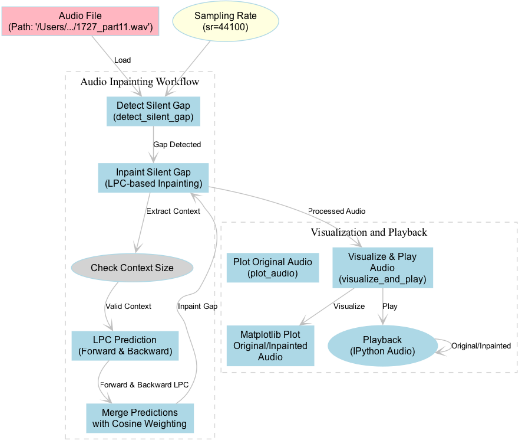

Figure 5: Audio Inpainting Model Workflow

4 Conclusion

In this work, we presented *Echoes Reborn*, an end-to-end neural pipeline for restoring historic piano recordings. The proposed system integrates three components—a classifier, a de-noising model, and an audio inpainter to address the challenges of gramophone noise, audio gaps, and high-frequency loss. Our classifier achieved a remarkable accuracy of 99.92% in identifying noisy versus clean audio samples, while the de-noising model improved the Signal-to-Noise Ratio (SNR) by 21.21 dB and achieved a Mean Absolute Error (MAE) of 0.0834. These results demonstrate the effectiveness of our approach in restoring high-fidelity audio from degraded recordings.

The visualization of noisy, clean, and de-noised spectrograms further highlighted the capability of the proposed system to preserve the integrity of musical details while mitigating noise. This work not only contributes to the preservation of cultural and musical heritage but also provides a foundation for further advancements in the field of audio restoration.

5 Future Work

While *Echoes Reborn* has demonstrated significant advancements, several areas for improvement remain. Inspired by recent research, we propose the following directions:

- **Advanced Architectures:** Incorporate diffusion-based models for audio restoration, as shown by Jean-Marie Lemercier [6], which balance interpretability and high performance for complex noise patterns.
- **Dataset Augmentation:** Expand datasets to include diverse instruments and noise types, following the approach of Ethan Manilow [7], who introduced MUSDB18-HQ for audio separation tasks.
- **Real-Time Processing:** Develop lightweight, low-latency models for real-time audio restoration, inspired by Alexandre Défossez [8], who optimized Demucs for dynamic audio processing.
- **Perceptual Metrics:** Integrate perceptual quality metrics like ViSQOL, as validated by Andrew Hines [9], to align system outputs with human auditory preferences.
- **Audio Inpainting:** Enhance generative methods for reconstructing missing audio, building on Andrés Marafioti [10], who employed GANs to effectively address audio gaps.

By exploring these advancements, we aim to enhance the robustness, scalability, and applicability of *Echoes Reborn* for real-world scenarios.

References

[1] Michael De Sapio (2024) Early Music and the Conservation of Culture. In, *The Imaginative Conservative*, Available at: https://theimaginativeconservative.org/2024/08/early-music-conservation-culture-michael-de-sapio.html.

[2] Takayuki, Sasaki. (1980) Sound restoration and temporal localization of noise in speech and music sounds. In, *Tohoku University*, Available at: https://psycnet.apa.org/record/1982-11282-001.

[3] Eloi Moliner, Maija Turunen, Filip Elvander & Vesa Välimäki. (2024) A Diffusion-Based Generative Equalizer for Music Restoration. In, *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx24), Guildford, United Kingdom*, Available at: https://arxiv.org/pdf/2403.18636.

[4] Eloi Moliner & Vesa Välimäki. (2022) A Two-Stage U-Net for High-Fidelity De-noising of Historical Recordings. In, *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Available at: https://ieeexplore.ieee.org/document/9746977.

[5] Andrés Marafioti, Nicki Holighaus, Piotr Majdak & Nathanaël Perraudin. (2022) Audio inpainting of music by means of neural networks. In *146th AES Convention*. Available at: https://arxiv.org/abs/1810.12138.

[6] Jean-Marie Lemercier, Julius Richter & Simon Welker. (2023) Diffusion Models for Audio Restoration. In, *Acoustics Lab, Department of Information and Communications Engineering, Aalto University, Espoo, Finland*, Available at: https://arxiv.org/pdf/2402.09821.

[7] Ethan Manilow, Gordon Wichern, Prem Seetharaman & Jonathan Le Roux. (2019) Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity. In, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Available at: 10.1109/WASPAA.2019.8937170

[8] Alexandre Défossez, Nicolas Usunier, Léon Bottou, Francis Bach. (2019) Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed. In, Cornell University, Available at: https://arxiv.org/abs/1909.01174

[9] Andrew Hines, Jan Skoglund, Anil Kokaram & Naomi Harte. (2015) "ViSQOL: An Objective Speech Quality Model." In, IWAENC 2012; International Workshop on Acoustic Signal Enhancement, Available at: https://ieeexplore.ieee.org/abstract/document/6309421.

[10] Andrés Marafioti, Nathanaël Perraudin, Nicki Holighaus & Piotr Majdak. (2019) A Context Encoder For Audio Inpainting. In, IEEE/ACM Transactions on Audio, Speech, and Language Processing , Available at: 10.1109/TASLP.2019.2947232.