Quality Enhancement of Highly Degraded Music Using Deep Learning-Based Prediction Models for Lost Frequencies

Arthur C. Serra
arthursrr@telemidia.puc-rio.br
TeleMídia/PUC-Rio
Brazil

Antonio José G. Busson
busson@telemidia.puc-rio.br
TeleMídia/PUC-Rio
Brazil

Álan L. V. Guedes
alan@telemidia.puc-rio.br
TeleMídia/PUC-Rio
Brazil

Sérgio Colcher
colcher@inf.puc-rio.br
Informatics Departament/PUC-Rio
Brazil

ABSTRACT

Audio quality degradation can have many causes. For musical applications, this fragmentation may lead to highly unpleasant experiences. Restoration algorithms may be employed to reconstruct missing parts of the audio in a similar way as for image reconstruction — in an approach called *audio inpainting*. Current state-of-the art methods for audio inpainting cover limited scenarios, with well-defined gap windows and little variety of musical genres. In this work, we propose a Deep-Learning-based (DL-based) method for audio inpainting accompanied by a dataset with random fragmentation conditions that approximate real impairment situations. The dataset was collected using tracks from different music genres to provide a good signal variability. Our best model improved the quality of all musical genres, obtaining an average of 12.9 dB of PSNR, although it worked better for musical genres in which acoustic instruments are predominant.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Audio quality enhancement, Audio reconstruction, Neural Networks, Autoencoder

1 INTRODUCTION

Audio quality degradation can have many causes. Signals can be disturbed by noise, or information can be corrupted by packet losses during transmission (*e.g.* voice-over-IP transmission), audio capture devices may exhibit many kinds of malfunctioning as well physical media may sometimes be partially damaged [**?**]. While for some applications short gaps in audio might be acceptable, for music applications this fragmentation may lead to highly unpleasant experiences. Restoration algorithms may be employed to reconstruct missing parts of the audio in a similar way as for image reconstruction. Adler *et. al.*[**?**] were the first to address this problem as an analogous to the image inpainting task — an approach called *audio inpainting*.

State-of-the-art methods for audio inpainting are usually based on Deep Learning (DL). Some of these methods are applied to the task of speech [**? ? ?**] or music [**? ? ?**] quality enhancement. However, such works cover limited scenarios, with well-defined gap windows and little variety of musical genres. This work investigates the application of benchmarking DL models for audio inpainting accompanied by a music dataset with random fragmentation conditions that approximate real impairment situations.

Given a fragmented audio signal, our model can predict the audio frequencies that were lost. Unlike in other methods, in our approach, we can predict frequencies in gaps of variable-sized windows and in random positions. Another contribution of this work is the construction of a dataset for the music quality enhancement task with many advantages over existing ones, as it has a more significant amount of music (13,583 tracks), and exhibits a greater variety of musical genres. Additionally, we analyzed the performance of DL models in each of the musical genres.

The remainder of this paper is organized as follows. Section 2 summarizes how recent work has been successfully applying DL-based methods in order to increase audio quality. Next, Section 3 describes the construction of our dataset. In Section 4 we introduce our proposal that incorporates the DL model to recover lost frequencies from fragmented audio signals, followed by Section 5 where we describe the experiments conducted to evaluate the effectiveness of our proposal. Section 6 is devoted to our final remarks and conclusions.

2 RELATED WORK

Adler *et al.* [**?**] were the first to define the audio reconstruction task as an audio inpainting problem, analogous to the image inpainting task. They define the reconstruction as the inverse problem of overlapping time-domain frames. Each inverse problem is solved using a sparse representation with the *Orthogonal Matching Pursuit algorithm* and the discrete cosine or Garbor dictionary. Although they only use linear methods for reconstruction, their work has brought a significant contribution by defining a previously unexplored problem.

Recent work investigating audio enhancement employs deep-learning-based such as Autoencoders [**?**], Generative Adversarial Networks (GANs) [**?**], and Long-Short Term Memory (LSTM) [**?**].

Lim *et al.* [**?**] present a super-resolution method for spectrogram band quality enhancement. They considered that the autoencoder strategy could achieve a satisfactory result using the frequency

and time domain representations. The autoencoder, called Time-Frequency Network (TFNet), has a branch for each representation, which in the last layer are merged into a high-resolution signal.

Marafioti *et al.* [**?**] focuses on temporal gaps reconstructions of audio with a fixed duration of 64 milliseconds. They built a controlled environment to demonstrate that the context associated with the lost signal facilitates the reconstruction process. Their dataset was composed exclusively of instrumental genre music, and the approach consisted of extracting features, such as linear prediction coding (LPC) and spectrogram. For each 320 ms, they applied a 64 ms gap in the center of the interval. Thereafter, a deep convolutional neural network acts as a context encoder to complete the produced central gaps. Finally, the reconstructed tracks were evaluated using the Objective Difference Grades (ODG) metric [**?**] that measures the human perception of the reconstruction.

Subsequently, Marafioti *et al.* [**?**] presented a GAN-based audio painting strategy for restoring long temporal gaps in music tracks. Their solution, called GACELA, considers two main aspects for reconstruction. Firstly, it determines five parallel discriminators to evaluate the reconstruction at five different context scales of the central gap. Secondly, it evaluates each context to determine the latent variables of the conditional GAN. They performed tests with gaps ranging from 320 ms to 1,500 ms, still centrally positioned in a larger context. However, they conclude that the artifacts generated during the reconstruction process remain noticeable.

Ebner *et al.* [**?**] presented a GAN-based reconstruction strategy for working with long gaps up to 500 ms. In their approach, gaps need to be centered in a specific period larger than the region to be repaired. They propose a Wasserstein GAN [**?**] with two discriminators, to evaluate a short and a long context prediction respectively. Then, these short and long context predictions are merged in an attempt to generate as little noise as possible for the listener. To demonstrate their results, they used instrumental music tracks from popular datasets such as MAESTRO [**?**] and evaluate the reconstruction using the ODG perception metric.

Morrone *et al.* [**?**] showed that audio inpainting tasks can also be performed in a multimodal way. They use both audio and video features concatenated frame-by-frame as inputs to a stacked Bi-directional LSTM (BLSTM). Their dataset contains a controlled content of an actor speaking to a camera positioned in front of him. Although the gaps to be filled in this strategy were randomly positioned, it is essential to point out that during the reconstruction process, the positions were known. They showed that when the gaps are too long, audio features are not enough for reconstruction, requiring the injection of visual features.

Unlike previous work, we propose an audio inpainting strategy that does not know gap positions. Moreover, by using autoencoders, our approach aims to generalize the inpainting process across different music genres.

3 DATASET

In this work, we use a part of the Free Music Archive (FMA) dataset [**?**] and adapt it for the audio reconstruction task. FMA is a large-scale dataset for evaluating several tasks in Music Information Retrieval (MIR). It provides full-length and high-quality audio, permissive license, pre-computed features, together with track- and user-level metadata, tags, and free-form text such as biographies. It consists of 343 days of audio from 106,574 tracks from 16,341 artists and 14,854 albums arranged in a hierarchical taxonomy of 161 genres.

To compose our dataset, we selected an FMA subset with 13,583 tracks distributed across 16 musical genres. Following the study done in [**?**], we selected the eight most representative musical genres to compose the training, validation, and part of the test set: Electronic, Experimental, Rock, Hip-Hop, Folk, Instrumental, Pop, and International. Additionally, we selected eight music genres in order to measure the model's generability: Classical, Historic, Jazz, Country, Soul-RnB, Spoken, Blues, and Easy Listening. Table 1 describes the distribution of each musical genre in the training, validation, and test sets.

Table 1: Distribution of musical genres in the training, validation and test sets.

Genre	Audio Qtdy	Train.	Valid.	Test
Electronic	1637	800	200	637
Experimental	1624	800	200	624
Rock	1608	800	200	608
Hip-Hop	1585	800	200	585
Folk	1518	800	200	518
Instrumental	1349	800	200	349
Pop	1186	800	200	186
International	1018	800	200	18
Classical	619	0	0	619
Historic	510	0	0	510
Jazz	384	0	0	384
Country	178	0	0	178
Soul-RnB	154	0	0	154
Spoken	118	0	0	118
Blues	74	0	0	74
Easy Listening	21	0	0	21

We have reduced the audio sampling rate to facilitate the task of predicting frequencies. Originally, FMA dataset provides the audios in the MP3 stereo format of 44 kHz and bitrate 320 kbps. We converted the selected audio to the WAV mono format of 16 kHz and bitrate of 256 kbps. Next, we extracted the spectrogram from each audio using the Short-Time Fast Fourier (STFT) algorithm of the Tensorflow library.[1] This process generates spectrograms with dimensions of 7500×128 for each audio. The computational power to train any model with these input dimensions is very high. So, we have split each spectrogram into patches of size 128×128, which represent 512 ms of the original audio. Finally, to generate the input for the prediction models, we perform a process to create random gaps, illustrated in Figure 1.

To create the gaps, we delete audio frequencies in windows of random sizes between 10% and 70% of the size of each patch. It was done in three steps: (1) The spectrogram is extracted from the audio using the STFT; (2) The spectrogram is split into patches of common size, these patches correspond to the set Y (ground truth);

[1] https://www.tensorflow.org/api_docs/python/tf/signal/stft

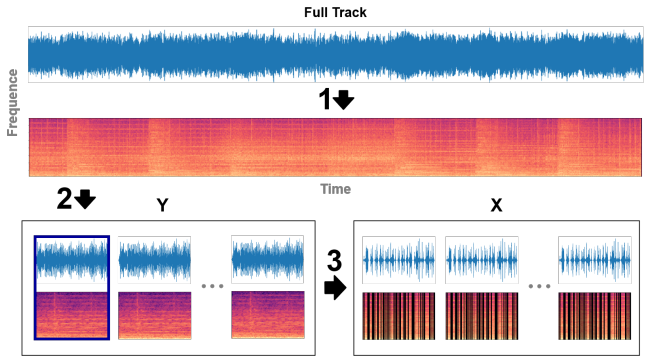

Figure 1: Process of creating our dataset.

(3) Set X (input) is generated with the process of deleting audio frequencies in windows of different positions and sizes.

Our dataset has some advantages over the previously published ones, such as: (a) the amount of data, with 13,583 tracks; (b) the variety of musical genres; (c) the multiple gaps with windows in different positions and sizes; (d) dimensional time patches equivalent to 520 ms.

4 METHOD

Our proposal relies on a DL model that learns how to recover lost frequencies from fragmented audio waves. Figure 2 illustrates the workflow of our method. Given a fragmented audio wave, the STFT is used to extract the corresponding spectrogram. Next, a DL model is used to predict the lost frequencies and restore the spectrogram. Finally, the Griffin-Lim [**?**] algorithm is used to convert the restored spectrogram back to the audio wave format.

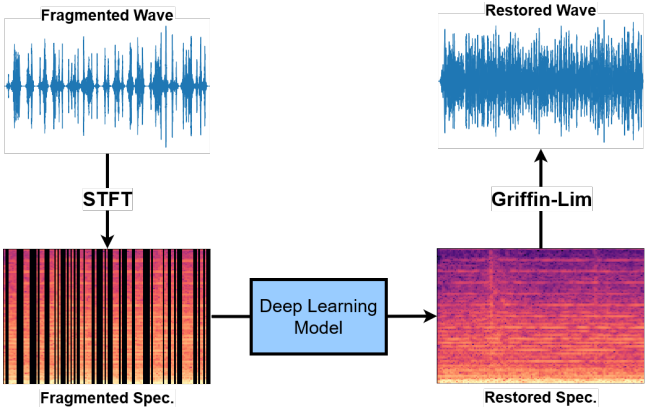

Figure 2: Audio reconstruction method.

At the heart of our method lies the DL-Model, over which we have focused much of our efforts. In the remainder of this section, we detail the different DL networks we tested as possible backbones to our DL-based method. Among so many other DL models in the literature, these were the ones that worked particularly well in our experiments described in Section 5.

4.1 U-Net

Figure 3 illustrates the U-Net [**?**] version used in this work. It is structured in 11 layers of convolutional blocks, where each block is a sequence of a convolution layer, with kernel size 3×3, a batch normalization, and a rectified linear units (ReLU) activation function. In the first five layers, the downsample process occurs, in which convolutional blocks are interleaved with max-pooling layers, such as kernel size 3×3 and stride 2. The upsampling process starts from the seventh convolutional block, and relies on a transposed convolution (deconvolution) layer, with kernel size 3×3 and stride 2. Each transposed convolution is concatenated with the corresponding downsample block output before moving on to the next upsample block. In addition, we also experimented with a variant of U-Net, called U-Net V2, which uses convolution layers with stride 2 in place of the max-pooling layer.

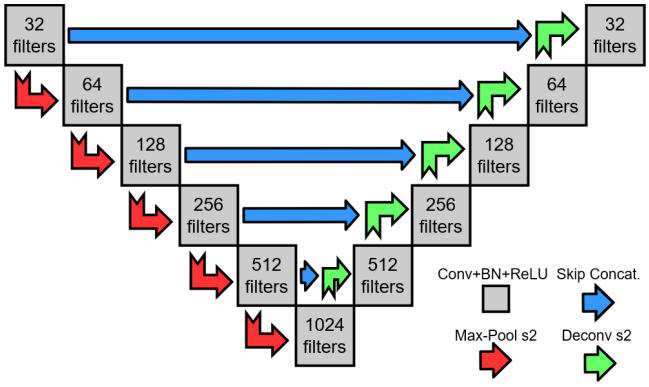

Figure 3: U-Net architecture

4.2 Res-U-Net

Figure 4 illustrates the Res-U-Net [**?**] architecture, in which the authors have extended the plain U-Net with the addition of a *Global Residual Connection* (GRC). The GRC mechanism is a trend in deep learning models for several image-to-image tasks, such as super-resolution, denoising, and artifacts removal [**? ?**]. The output of the U-net is connected to a last convolutional layer with three filters, kernel size 3×3, and linear activation to produce the residual features. Then, it is subtracted from the input by the GRC to reconstruct the output. We also experimented with a version of Res-U-Net that uses U-Net V2 as a backbone.

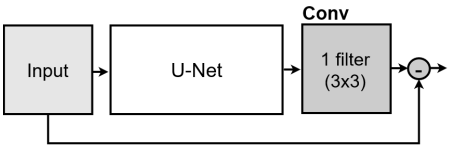

Figure 4: Res-U-Net architecture.

4.3 Attention U-Net

Figure 5 illustrates the Attention-U-Net [**?**] architecture, a variation of U-Net combined with a spatial attention mechanism. The attention block has two inputs, a feature map coming from the previous layer (Skip) and the other from the upsampling (Up) operation by deconvolution. Convolutional layers use both feature maps with N filters, kernel size 1×1, and linear activation, where N is the number of channels of the input map features. Then, they are added together and activated by the ReLU function. Finally, the output of a convolutional layer with sigmoid activation is multiplied with the result of the convolution of the skip connection to generate the output of the attention block.

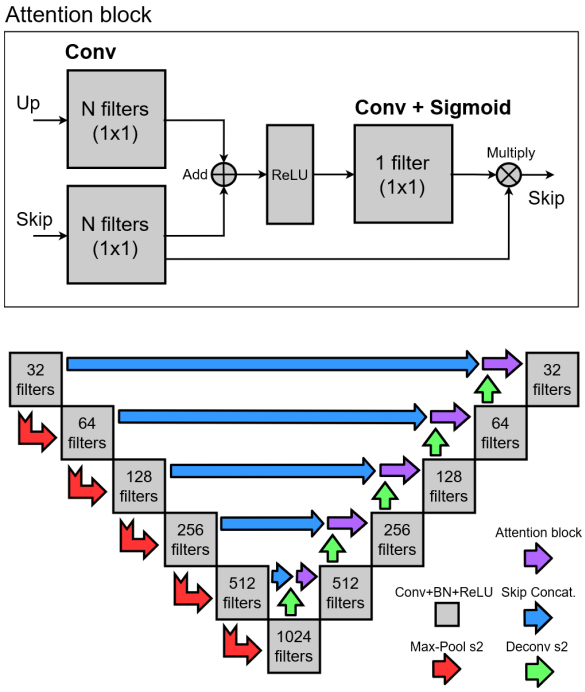

Figure 5: Attention U-Net architecture.

5 EXPERIMENT

In this section, we describe our models and evaluate their effectiveness in restoring spectrograms. It is worth pointing out that our implementations of the networks, training module, saved models, and datasets can be obtained in our public git repository.[2].

The remainder of this session is structured as follows. Subsection 5.1 presents the metrics to measure the effectiveness of the models. Next, Subsection 5.2 presents the setup of the experiment. Finally, Subsection 5.3 contains our empirical findings.

5.1 Metrics

We evaluate our models using three different metrics: the PSNR (Peak Signal-to-Noise Ratio), the NRMSE (Normalized Root Mean Square Error), and the ODG (Objective Difference Grade). We use these metrics to evaluate two different aspects; more precisely, we use the PSNR and the NRMSE to measure the quality of the spectrogram restoration, while we use the ODG to measure the human-perceivable noise.

5.1.1 PSNR. The PSNR is a popular metric used to measure the quality of signal restoration [**?**]. It is a logarithmic measure (in decibels) of the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. More formally, given two images x and y of size $m \times n$, the PSNR is calculated by:

$$PSNR(x, y) = 10 \log_{10}\left(\frac{MAX^2}{MSE(x, y)}\right) \tag{1}$$

where MAX is the maximum possible pixel value of the image, and

$$MSE(x, y) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [x(i, j) - y(i, j)]^2 \tag{2}$$

5.1.2 NRMSE. The NRMSE is another popular metric to measure the quality of signal reconstruction. This metric consists of the square root of MSE normalized by the range (defined as the maximum value minus the minimum value) of the measured data. Given two images x and y of a common size, the NRMSE is defined as

$$NRMSE(x, y) = \frac{\sqrt{MSE(x, y)}}{y_{max} - y_{min}} \tag{3}$$

with the MSE defined by (2).

5.1.3 ODG. The Perceptual Evaluation of Audio Quality (PEAQ) is a standardized algorithm to measure audio reconstruction quality objectively. Its output is the *Objective Difference Grade* (ODG). The algorithm calculates various Model Output Variables (MOV), consisting of a vector with 11 features based on the human psychoacoustic model. We use the model created by Kabal *et al.* [**?**], which is based on the ITU-R BS.1387 standard.[3]

Figure 6 illustrates the PEAQ algorithm, given the reference and test signals, the psycho-acoustic model generates the MOV. Next, the cognitive model aggregates the MOV features to predict the distortion index D_I.

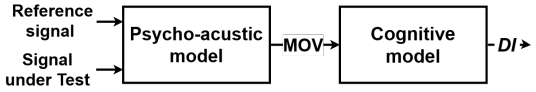

Figure 6: Stages of PEAQ.

Finally, the ODG is calculated by the following equation:

$$ODG = b_{min} + (b_{max} - b_{min}) sig(D_I) \tag{4}$$

where sig represents the sigmoid function, parameters $b_{min} = -3.98$ and $b_{max} = 0.22$.

Table 4 shows how to interpret the ODG values. The scale goes from -4 to 0. The closer to 0, the smaller the difference in perception between the reference signal and the signal under test.

[2]https://github.com/TeleMidia/audio_reconstruction

[3]https://www.itu.int/rec/R-REC-BS.1387

Table 2: PSNR Results of models on validation set. ▲

Genre	U-Net	Attention U-Net	U-Net V2	Res-U-Net	Attention Res-U-Net	Res-U-Net V2
Instrumental	59.0135	59.0354	59.5594	59.1736	59.1364	**59.7195**
Folk	58.0909	58.1038	58.6509	58.2547	58.2277	**58.8289**
International	57.5399	57.5298	58.0458	57.7538	57.7006	**58.1821**
Experimental	57.3531	57.3476	57.7869	57.5243	57.4582	**57.8991**
Pop	54.9014	54.8721	55.3386	54.9985	54.9390	**55.4537**
Electronic	53.6760	53.6068	54.1059	53.7744	53.6746	**54.2169**
Hip-Hop	52.9716	52.8735	53.3412	53.0586	52.9872	**53.4567**
Rock	52.6030	52.5724	52.9829	52.6776	52.5934	**53.0700**
Mean	55.7687	55.7427	56.2264	55.9020	55.8396	**56.3534**

Table 3: NRMSE Results of models on validation set. ▼

Genre	U-Net	Attenion U-Net	U-Net V2	Res-U-Net	Attention Res-U-Net	Res-U-Net V2
Folk	0.1395	0.1398	0.1326	0.1378	0.1387	**0.1313**
International	0.1420	0.1430	0.1361	0.1403	0.1416	**0.1356**
Instrumental	0.1454	0.1456	0.1386	0.1438	0.1449	**0.1374**
Electronic	0.1540	0.1560	0.1488	0.1534	0.1556	**0.1486**
Hip-Hop	0.1531	0.1556	0.1485	0.1527	0.1548	**0.1486**
Pop	0.1556	0.1566	0.1499	0.1547	0.1561	**0.1495**
Experimental	0.1579	0.1586	0.1524	0.1563	0.1579	**0.1520**
Rock	0.1705	0.1717	0.1651	0.1697	0.1718	**0.1650**
Mean	0.1523	0.1534	0.1465	0.1511	0.1527	**0.1460**

Table 4: ODG interpretation.

ODG	Impairment
0	Imperceptible
-1	Perceptible, but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very Annoying

5.2 Setup

Our networks were trained using an octa-core i7 3.40 GHz CPU with a NVIDIA TESLA K80 GPU. The training was based on the Adam [**?**] optimization with the momentum of 0.999, exponential decay of 0.9 and epsilon of 1e-07, batch normalization with the decay of 0.9997 and epsilon of 0.001 with a fixed learning rate of 0.001, and MSE (Mean Square Error) as the loss function. The network weights were initialized with Glorot [**?**] and seed zero. We normalized our dataset and ran our experiments for a maximum of 100 epochs.

5.3 Results

5.3.1 Validation. Tables 2 and 3 show the results according to the metrics PSNR and NRMSE for the experiment on the validation set. For all musical genres, the best result was achieved by the Res-U-Net V2, which produced a mean PSNR of 56.3534 dB and a mean NRMSE of 0.1460, followed by the U-Net V2, Res-U-Net, Attention Res-U-Net, U-Net, and Attention U-Net models. Among the musical genres chosen to compose the dataset, some genres have a higher correlation with others in our dataset. The Instrumental genre, for example, has characteristics similar to the Folk and International genres, as they are genres with a more significant presence of notes made by acoustic instruments. On the other hand, Electronic, Hip-Hop, and Rock have a large number of notes produced by electronic instruments and sound synthesizers. Acoustic instruments produce more uniform audio waves, while electronic instruments tend to produce more variable audio waves. We observed that our model better predicts uniform wave frequencies. Therefore, the quality of the reconstructed audio is better in the genres that have notes produced by acoustic instruments. Table 2 shows that the model produced by Res-U-Net V2 has better performance in the Instrumental, Folk, and International genres and the worst result in the Pop, Electronic, Hip-Hop, and Rock genres.

Figure 7 shows the training convergence curve of the top-3 models. Note that their curves are close. When analyzing the convergence curve of the PSNR metric, the Res-U-Net V2 network converged faster and remained above the others during all training epochs. The best model produced by the Res-U-Net network was obtained at epoch 92. Another point that is worth mentioning is that the convergence curve of the Res-U-Net network is more unstable than the others. From the perspective of deep learning engineering, among the mechanisms used to extend the U-Net vanilla, the ones that increased performance were the GRC and the replacement of max-pooling by the convolutional layer with stride 2. The Res-U-Net V2, which uses both mechanisms, achieved better results than Res-U-Net (using only GRC) and U-Net v2 (using only the convolutional layer with stride 2).

Although the spatial attention mechanism is a trend in the deep learning field, networks extended with a spatial-attention mechanism obtained poor performance in this experiment. Spatial-attention convolutions rely on dense information, but the nature of

the spectrogram information is sparse. Zheng *et al.* [**?**] solved this problem by combining temporal self-attention and frequency-wise self-attention parallelly for capturing global dependency along temporal and frequency dimensions in a separable way. In the future, we will experiment with this method of attention mechanism to attest if it is effective in the context of this work.

5.3.2 Test. The results of the best model for the test sets are summarized in Table 5. In detail, for each musical genre, is described the PSNR, NRMSE, and ODG corresponding to the before, after of the restoration of its spectrograms. On average, the model produced by the Res-U-Net V2 network achieved gains of approximately 12.9 dB of PSNR, -0.42 of NRMSE, and 1.56 of ODG. The result shows that the genres that are only in the test set (Classical, Historic, Jazz, Country, Soul-RnB, Spoken, Blues, and Easy Listening) had a similar gain to the other musical genres that were used only for testing, attesting to the generability power of the model. The full demonstration of our model is available on Youtube[4] (temporary URL due to blind-review).

6 FINAL REMARKS

In this work, we proposed a DL-based method to enhance the quality of highly degraded music files. The core of our method consists of an autoencoder that learns how to recover lost frequencies from fragmented audio waves. For that, we first created a dataset based on the FMA. Our dataset has advantages over the existing and previously published ones, mainly due to the number of tracks, the variety of musical genres, the variety in the position and size of the gap windows. Next, we experimented with the produced datasets to choose the best model based on the ODG, PSNR, and NRMSE metrics. Among the 7 DL networks analyzed, the Res-U-Net V2 network obtained the best performance.

Our model improved the quality of all musical genres, obtaining an average of 12.9 dB of PSNR. As noted in the experiments, our model works a little better for musical genres in which acoustic instruments are predominant than for genres where electronic instruments are more present.

Although our model improves the quality of songs, residual noise is still noticeable. Aiming for ODG results closer to 0 for all musical genres, in the future, we plan to experiment with others methods, such as double stage models based on cascade models [**?**], WaveNet vocoder [**?**], and sparse transformers [**?**].

We also observed that networks extended with a spatial-attention mechanisms obtained poor performance in our experiment. We plan to combine temporal self-attention and frequency-wise self-attention in parallel for capturing global dependency along temporal and frequency dimensions in a separable way [**?**].

ACKNOWLEDGMENTS

This material is based upon work supported by Air Force Office Scientific Research under award number FA9550-19-1-0020.

[4] https://streamable.com/gzg1d4

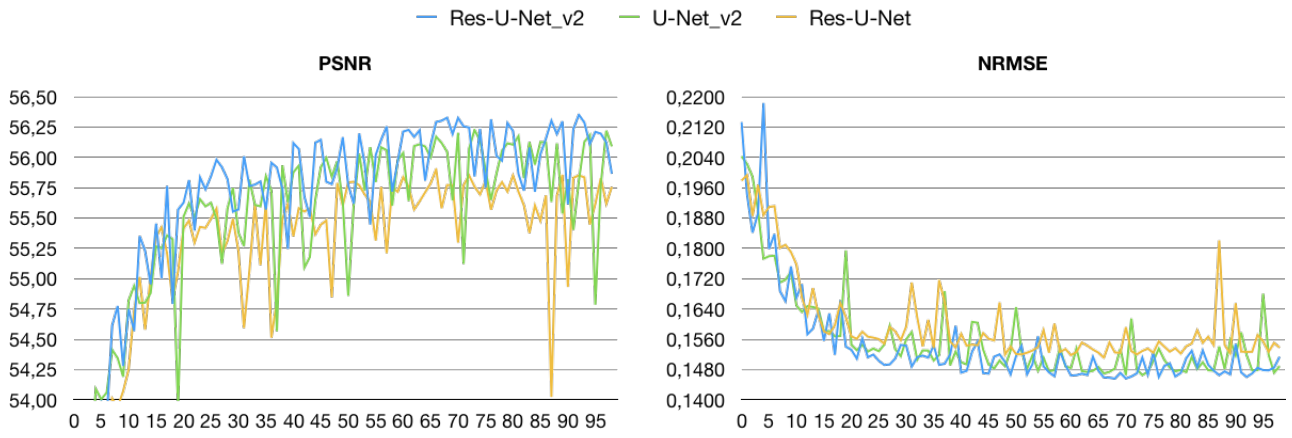

Figure 7: Convergence curve of the top-3 models.

Table 5: Results of Res-U-Net V2 on test set

	Before			After			Gain		
Genre	**PSNR▲**	**NRMSE▼**	**ODG▲**	**PSNR▲**	**NRMSE▼**	**ODG▲**	**PSNR▲**	**NRMSE▼**	**ODG▲**
International	48.7529	0.5300	-3	61.9354	0.1239	-1	+13.1825	-0.4061	+2
Blues	45.2932	0.5611	-3	58.2071	0.1403	-1	+12.9138	-0.4208	+2
Easy Listening	44.3598	0.5655	-3	57.4870	0.1386	-1	+13.1272	-0.4269	+2
Pop	43.3860	0.5770	-3	55.9381	0.1511	-1	+12.5520	-0.4258	+2
Country	42.9498	0.5784	-3	55.5137	0.1500	-1	+12.5638	-0.4284	+2
Electronic	41.4648	0.5845	-3	54.3265	0.1512	-1	+12.8617	-0.4333	+2
Hip-Hop	40.8066	0.5846	-3	53.7808	0.1489	-1	+12.9741	-0.4357	+2
Rock	41.8011	0.5831	-3	53.5114	0.1660	-1	+11.7103	-0.4170	+2
Soul-RnB	40.7757	0.5883	-3	53.4086	0.1552	-1	+12.6329	-0.4331	+2
Classical	55.1747	0.4930	-3	68.5498	0.1112	-2	+13.3751	-0.3818	+1
Spoken	52.3546	0.5097	-3	65.0444	0.1231	-2	+12.6897	-0.3866	+1
Historic	49.4550	0.5548	-3	64.1591	0.1146	-2	+14.7040	-0.4402	+1
Jazz	46.9471	0.5584	-3	60.3281	0.1329	-2	+13.3810	-0.4255	+1
Folk	45.7114	0.5702	-3	59.4140	0.1306	-2	+13.7026	-0.4396	+1
Instrumental	45.6390	0.5687	-3	58.6380	0.1430	-2	+12.9989	-0.4256	+1
Experimental	46.1910	0.5540	-3	58.3694	0.1521	-2	+12.1783	-0.4019	+1
Mean	45.6914	0.5601	-3	**58.6632**	**0.1395**	**-1.47**	**+12.9717**	**-0.4205**	**+1.56**