# HIGH FREQUENCY MAGNITUDE SPECTROGRAM RECONSTRUCTION FOR MUSIC MIXTURES USING CONVOLUTIONAL AUTOENCODERS

*Marius Miron* *

Independent researcher
miron.marius@gmail.com

*Matthew E.P. Davies*

INESC TEC
Sound and Music Computing Group
Porto, Portugal
mdavies@inesctec.pt

## ABSTRACT

We present a new approach for audio bandwidth extension for music signals using convolutional neural networks (CNNs). Inspired by the concept of inpainting from the field of image processing, we seek to reconstruct the high-frequency region (*i.e.*, above a cutoff frequency) of a time-frequency representation given the observation of a band-limited version. We then invert this reconstructed time-frequency representation using the phase information from the band-limited input to provide an enhanced musical output. We contrast the performance of two musically adapted CNN architectures which are trained separately using the STFT and the invertible CQT. Through our evaluation, we demonstrate that the CQT, with its logarithmic frequency spacing, provides better reconstruction performance as measured by the signal to distortion ratio.

## 1. INTRODUCTION

Audio signals are often low-passed, encoded or compressed before transmitting them through phone lines and Internet streams. This results in the loss of high frequency content and compromises audio quality. Narrow-band audio signals which have information up to a certain frequency cutoff can be perceptually enhanced by reconstructing the higher frequency content. This research task, known as *audio bandwidth extension*, attempts to increase the perceived or real frequency spectrum of audio signals [1, 2, 3, 4, 5].

Audio bandwidth extension methods have been applied to speech signals in an unsupervised and supervised manner. The former are typically statistical approaches which model the relationship between low and high frequency spectral content by relating lower and upper harmonics [1]. For instance, the linear predictive coding (LPC) method in [2] analyzes the lower frequency spectra to synthesize high frequency components. It relies on a codebook: a dictionary of wide-band envelopes, which are matched with the envelope of narrow-band spectral frames. Spectral band replication [6] on the other hand transposes up harmonics from lower and midrange frequencies to higher bands.

Supervised methods learn priors from wide-band signals which are later used to recover the high frequency content of narrow-band signals. Matrix decomposition methods such as non-negative matrix factorization (NMF) [3, 5] treat the magnitude spectrogram as combinations of priors in the form of non-negative bases. At the test stage, these bases are kept fixed and are used to estimate the NMF parameters which best explain the narrow-band signal.

Methods using neural networks learn priors from features derived from time-frequency representations to predict high-band spectral envelopes [7, 8]. Bandwidth extension with deep neural networks has been shown to increase the robustness of speech recognition [8]. In addition, the resolution of raw audio signals, regarded as time series, can be increased using convolutional neural networks (CNNs) [9].

In this paper we seek to estimate high frequency components in time-frequency representations of music signals. Compared to speech, music signals are often complex mixtures, comprising a variety of instruments, both percussive and harmonic, singing voice, and non-linear audio effects. Thus, music signals have broader, richer, and perceptually more relevant high frequency content, which is therefore more difficult to estimate.

While the aim of bandwidth extension for speech is tightly coupled with signal compression and band-limited communication channels, for music signals there are important distinctions both in terms of the constraints of the problem and the potential applications. First and foremost, our aim is to perform bandwidth extension up to CD quality (*i.e.*, 44.1 kHz sampling rate with a Nyquist rate of 22.05 kHz). Given the absence of harmonic information in high frequency musical content (*e.g.*, above 10 kHz), our proposed musical bandwidth extension will be required to reconstruct percussive-type content. Depending on the bandwidth of the narrow-band input signal, it may also be required to reconstruct the upper partials of harmonic content present in the narrow-band signal. In this way, perceptually accurate musical bandwidth extension could be used to replace high-band information typically lost via lossy compression in audio formats such as MP3 and AAC, and thus reduce the bandwidth overhead when streaming music, or allocate a higher bit rate for lower frequency information.

Our specific long term goal is to explore a more creative application of audio bandwidth extension, namely towards the restoration of old music recordings. To this end, we seek to renew old recordings (in particular, jazz from the 1940s and 50s) and thus allow modern-day listeners to experience this music in high audio quality as performed by the original musicians. Towards this ambitious goal, we first investigate the feasibility of full-bandwidth extension for music signals under more controlled conditions, which can be more readily evaluated via access to both the full- and band-limited versions.

Similar to the concept of image inpainting or completion [10, 11], for which CNNs have been shown to be particularly adept, we aim to learn localized features in order to recover the missing higher frequency regions of short-term Fourier transform (STFT) and constant-Q transform (CQT) stereo magnitude spectrograms [12]. However, since the time and frequency axes in STFT and CQT representations do not correlate in the same way

* Marius Miron is currently a post-doctoral researcher at the European Commission Joint Research Center

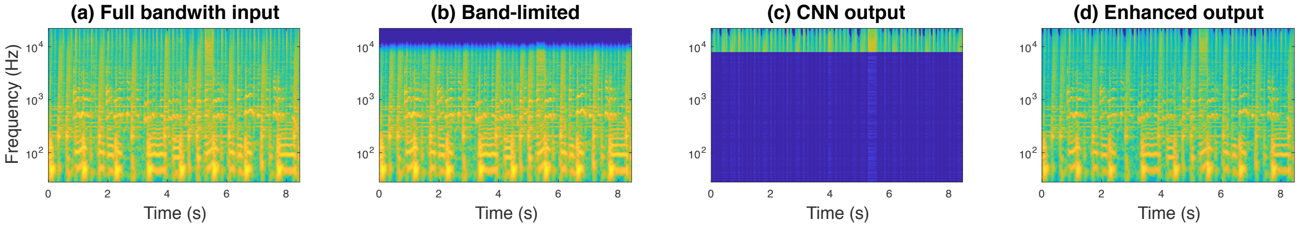

Figure 1: *Illustrative overview of our proposed approach for bandwidth extension. (a) The CQT of a short musical audio input sampled at* 44.1 *kHz. (b) The band-limited version resulting from a low-pass filter with a cutoff frequency of* 7500 *Hz. (c) The high frequency output of the CNN*[1]*. (d) The enhanced output signal obtained by combining the band-limited and CNN reconstruction.*

as the axes of an image, we explore two musically motivated CNN architectures: bottleneck and stride [13, 14] rather than more standard square filters in image processing.

For our musical inpainting problem, we aim to reconstruct or "complete" a strip covering the highest frequency bins of a time-frequency, for which an illustrative example is shown in Figure 1. While this is conceptually related to the idea of filling temporal gaps (*i.e.*, missing vertical strips) [15, 16] these methods exploit temporal redundancy via repetition in the musical input, where as in our approach, the high frequency region is never observed.

A particular novelty of our proposed approach is to leverage implicit knowledge of musical structure by the use of the constant-Q spectrogram. For bandwidth extension, the CQT has a potentially advantageous property over the STFT, which is that, due to the logarithmic spacing of the CQT bins, we can make a richer observation of the narrow-band (*i.e.*, low-frequency) region in order to reconstruct a smaller amount of higher frequency information. Comparing the STFT and CQT in matrix form (where rows correspond to frequency and the columns to time) this means that for an identical cut-off frequency (*e.g.*, of $f_s/4$), and a roughly equal total number of frequency channels, a far smaller amount of data must be reconstructed for the CQT than for the STFT. Until recently, such potential benefits remained theoretical due to the absence of an inverse CQT transform. However, recent work leveraging the non-stationary Gabor transform (NSGT) [17, 18] has demonstrated that perfect reconstruction of the CQT is both possible and executable in reasonable computation time.

For this initial work, our primary focus is towards the reconstruction of magnitude spectrograms, thus we do not attempt any automatic reconstruction of the phase spectrogram. Instead we make use of the original phase from the band-limited version, without any subsequent modification. Our evaluation focuses on the measurement of the signal to distortion ratio (SDR) for the enhanced and band-limited versions. In this way, the extent of the enhancement provided by our approach can be assessed by the increase in SDR over the band-limited versions.

The remainder of this paper is structured as follows. In Section 2 we contrast our approach with existing work in audio bandwidth extension. In Section 3, we detail our proposed method using convolutional neural networks, which we evaluate in Section 4, and provide discussion and conclusions in Section 5.

## 2. RELATION WITH PREVIOUS WORK

With the exception of [5, 9, 19], most previous research in audio bandwidth extension has been applied to speech signals. Regarding the methodology, the deep learning approaches in [8, 9] are the closest to our proposed method. In the same way as [9], which uses a similar approach to image super-resolution [20], we are inspired by recent advancements in image processing using CNNs [10, 11]. Unlike [7] we eliminate all accompanying heuristics and estimate the high-frequency spectra directly with the neural networks.

In contrast to the NMF speaker-specific spectral bases used in [3, 19] or the codebook of the LPC approach [2], we are concerned with the generalization capabilities of our trained model and do not seek to tailor our approach for specific individual pieces of music. Furthermore, we do not tune any method-specific hyper-parameters or weighting coefficients which were previously used in [2] as a part of a chain of signal processing heuristics.

Similar to the convolutional NMF approach in [3], the hidden Markov models (HMM) in [21], and the time-series CNN in [9], we consider cross-frame contextual dependencies. These short-term dependencies are learned by CNNs using horizontal filters for a given time-context, while timbre features are learned using vertical filters [13, 14].

The CNN approaches used in image restoration, completion, or inpainting [22, 10, 11] are exposed to the entire image and not just to the missing patches in order to perform the reconstruction. In a similar fashion, we use the observation of the lower frequencies to better reconstruct the higher frequencies.

## 3. METHOD

### 3.1. Overview

An overview of our proposed method, which comprises two stages: training and enhancement, can be seen in Figure 2. For training we require a dataset comprising full-bandwidth music recordings and narrow-band versions which lack high frequency content above a specific cutoff frequency. We obtain narrow-band versions by applying a low-pass filter to the original recordings. Then, we compute the desired time-frequency representation, using the STFT or CQT, and extract the respective magnitude spectrogram for each channel of the stereo recordings. Additionally, we apply the data processing heuristics described in [23] and train the CNNs with the architectures described in Section 3.3 and the training procedure in Section 3.4.

The enhancement stage is detailed in the Section 3.5, where the high-frequency content is obtained by feeding the magnitude

[1] While the CNN outputs a full wide-band spectrogram, the region below the cut-off has been attenuated for greater visual clarity.

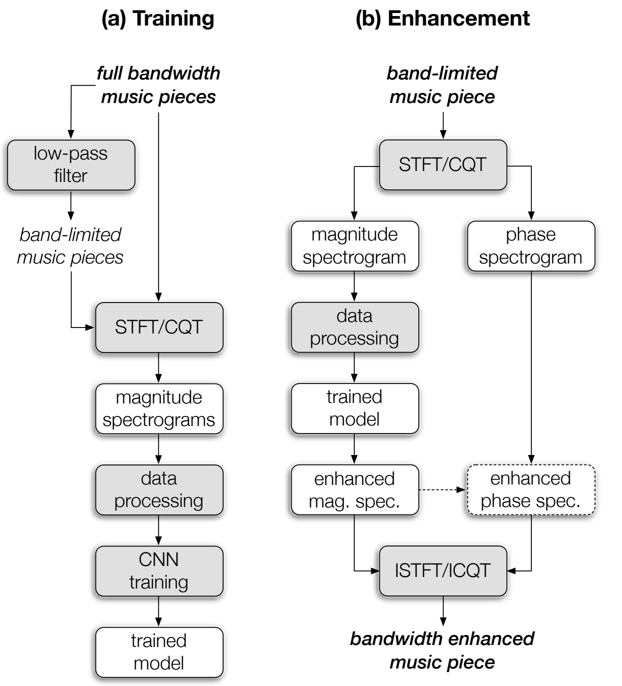

Figure 2: *Overview of our bandwidth extension system. (a) The training stage has access to full-band and band-limited music signals. (b) The enhancement stage only observes the band-limited signals. Boxes shaded in grey indicated processes, where as those in white correspond to data. The term data processing is used to encapsulate the partitioning of the data into overlapping chunks. The dashed arrow and box indicate optional processing which is not undertaken in this work.*

spectrograms forward through the previously trained CNN. The phase spectrogram of the band-limited version is retained to compute the inverse STFT or CQT.

## 3.2. Feature computation

We calculate the STFT or the CQT [18] of the stereo audio mixture as $\mathbf{X}_i(t, f)$ where $i = 1, 2$ are the stereo channels, $t$ is the time axis and $f$ is the frequency axis. In order to focus on the reconstruction of the magnitude spectrum, we discard the phase when computing the training features for the neural network.

The CNN architectures used in this paper require a fixed input size $(T, F)$, where $T$ is the temporal context in time frames and $F$ is the total number of frequency bins corresponding to the STFT or CQT magnitude spectrograms. To obtain magnitude spectrograms of fixed duration, the variable-size magnitude spectrograms of each music piece are split into overlapping chunks of fixed size $T$ time frames with an overlap of $O$ frames. In addition, splitting the input signal into chunks leads to a smaller network, with fewer parameters to train, and thus a lower computational burden. These data processing heuristics adopted prior to training are described in detail in [23] and were used previously for the task of audio source separation for full length musical recordings [14, 23, 24].

## 3.3. Convolutional autoencoders

We present two musically motivated CNN autoencoder architectures, the CNN bottleneck in Section 3.3.1 and the CNN stride-2 in Section 3.3.2. Since time and frequency in magnitude spectrograms have different meanings than the horizontal and vertical axes in images, we should not adopt image-processing square filters. Instead, we follow [13, 14] by using vertical filters to model frequency components and horizontal filters to model their temporal evolution.A further distinction is that the magnitude spectrograms of audio signals are sparse [25]. Thus, we use a sparse activation function between the layers, specifically, rectified linear units (ReLU) [26]. In addition, the CNN bottleneck architecture has a dense bottleneck layer with a low number of units to compress, or reduce, the learned features. On a related note, the CNN stride-2 architecture comprises successive convolutions with a stride[2] of two which is the equivalent of learning features by successively downsampling the inputs by a factor of two.

The inputs to both the CNN architectures are multiple magnitude spectrograms of size $(T, F)$, across the channel dimension $i$. In our case, the learned feature maps are shared between the two input channels [26]. We argue that the CNN can learn more diverse filters from music mixtures with a wide stereo image and therefore we provide magnitude spectrograms for both channels as input. In a further parallel with image processing, this can be considered similar to using the RGB layers of colour images rather a single greyscale image.

The CNN autoencoders comprise an encoding and a decoding stage. The encoding stage contains convolutional and feedforward layers, while the decoding stage performs the inverse operations of the convolutions in the reverse order such that the output of the CNN has the same dimensions as its input, $(2, T, F)$. Note, we do not use a soft-mask as in music source separation, but instead we directly estimate the magnitude spectrogram with enhanced high-frequency content, $\hat{\mathbf{X}}$. In addition, we assume that the frequency content to be recovered does not have higher energy than the low frequency content. To this end, we limit all the values of $\hat{\mathbf{X}}_i(t, f)$ to the maximum value in channel $i$ at time frame $t$ of the input $\mathbf{X}_i(t, f)$.

### 3.3.1. CNN bottleneck

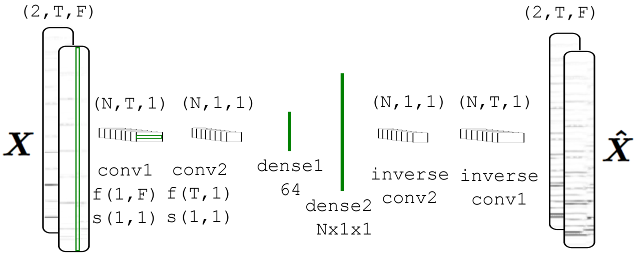

Figure 3: *CNN bottleneck autoencoder architecture [14]. For each layer we give the shape of the filters, strides and feature maps.*

We test a version of the CNN bottleneck successfully used in music source separation [14, 24, 23]. A diagram of the architecture is depicted in Figure 3, and comprises a horizontal convolution, *conv1*, a vertical convolution *conv2*, a bottleneck dense layer

[2] The stride controls how much a filter is shifted on the input.

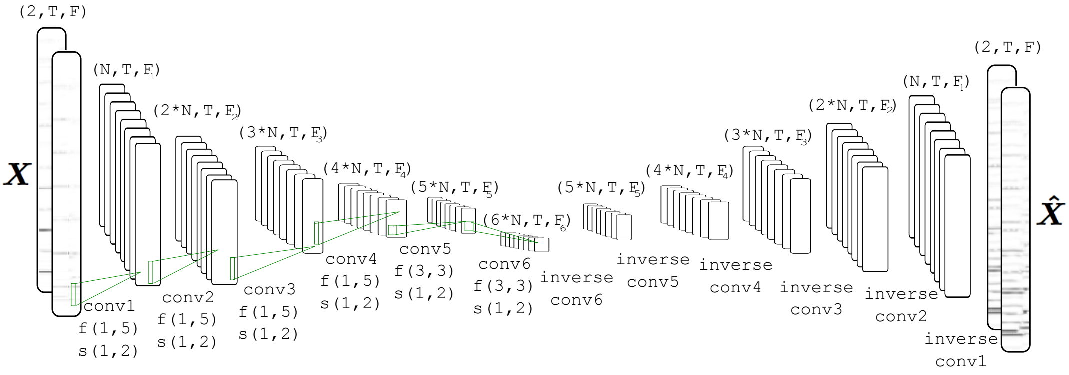

Figure 4: *CNN stride-2 autoencoder architecture. For each layer we give the shape of the filters, strides and feature maps.*

*dense1*, and another dense layer *dense2* to recover the dimensionality needed to perform the inverse operations of *conv2* and *conv1*. We have $N$ filters for $conv1$ and $conv2$.

#### 3.3.2. CNN stride-2

Small successive convolutional layers with a stride of two have been shown to reduce the number of parameters in a network [27]. Therefore, in contrast to the CNN bottleneck, we target a deep architecture comprising small convolutions. Moreover, time-frequency representations of musical signals often exhibit evenly spaced harmonic components. By modeling frequency content in strides of two we aim to capture high frequency harmonics learned from their low frequency counterparts.

An overview of the stride-2 architecture is shown in Figure 4. For each layer $k$, the feature maps reduce their frequency size: $F_k = (F_{k-1} - 5)/2 + 1$, as explained in [24]. We have four successive $(1, 5)$ convolutions in frequency, followed by two, two-dimensional $(3, 3)$ convolutions to capture the time-frequency dependencies, each considering the reduction performed by the previous layers.

### 3.4. Training procedure

Although the output of the CNN, $\hat{\mathbf{X}}$, contains a reconstruction of the magnitude spectrogram across all frequency bins, the parameters of the autoencoder are trained according to a loss function which only considers the reconstruction in higher frequencies. Thus, the loss function $L_c$ depends on the cutoff frequency in bins $c$ and is defined in equation (1) as the mean-squared error (MSE) between the target magnitude spectrogram $\bar{\mathbf{X}}$, and the estimated magnitude spectrograms, $\hat{\mathbf{X}}$:

$$L_c = \sum_{t,f,i} \|u(f-c)(\bar{\mathbf{X}}_i(t,f) - \hat{\mathbf{X}}_i(t,f))\|^2, \quad (1)$$

where $u(f-c)$ is the unit step function which is 0 for the bins lower than $c$ and 1 for the bins greater than or equal to $c$.

The parameters of the CNN are updated according to the loss function $L_c$ using mini-batch Stochastic Gradient Descent with the *Adamax* algorithm [28].

### 3.5. Enhancement

When computing the STFT or CQT for enhancement, we retain the phase and we split the magnitude spectrogram into overlapping chunks of size $T$ time frames with an overlap of $O$ frames as in the training stage. For each chunk $\mathbf{X}$ we obtain an estimation $\hat{\mathbf{X}}$. We then use the estimated chunks to reconstruct the enhanced magnitude spectrogram through the overlap-add procedure as described in [23] and as used in [14, 23, 24].

In contrast to deep learning source separation methods, the estimated spectrogram is not the result of Wiener filtering [29] which ensures that the spectrograms of the sources sum to the input spectrogram. Instead, we need to ensure that the original low-bandwidth content is preserved. To this end, we blend the high-frequency part of the estimations yielded by the network, $\hat{\mathbf{X}}$, with the low-frequency part of the input, $\mathbf{X}$:

$$\tilde{\mathbf{X}}_i(t,f) = (1 - r_c(f))\mathbf{X}_i(t,f) + r_c(f)\hat{\mathbf{X}}_i(t,f) \quad (2)$$

where $r_c(f) = \max(0, \min(1, f - c))$ is a ramp function depending on the the cutoff frequency in bins $c$.

As specified in Section 1, we only attempt to reconstruct the magnitude spectrum – without access to phase information when training. However, in order to invert either the reconstructed STFT or CQT we must provide phase information. To this end, we use the phase spectrogram from the band-limited version, as shown in Figure 1(b). Finally, the bandwidth extended audio signals are obtained using with an inverse overlap-add STFT or inverse CQT [18].

## 4. EVALUATION

The basis of our evaluation is to compare the reconstruction from the STFT and CQT, with the two different CNN autoencoder models: bottleneck and stride-2, and across two cutoff frequencies of 3500 Hz and 7500 Hz. In total, this creates eight reconstruction conditions for comparison.

### 4.1. Experimental setup

We test our approach on the publicly available Medleydb dataset [30] comprising 121 multi-tracks from which we use the stereo mixes (in uncompressed .wav format sampled at 44.1 kHz and with 16-bit resolution). The dataset covers the following genres: Singer/Songwriter, Classical, Rock, World/Folk, Fusion, Jazz, Pop, Musical Theatre, Rap. There are 52 instrumental tracks and 70 tracks containing vocals. We randomly split the dataset in training and testing subsets with a ratio of $0.8$ (*i.e.*, 80% for training and 20% for testing).

#### 4.1.1. Evaluation metrics

As the basis for the evaluation, we use the BSS_Eval framework [31], a widely used tool to objectively evaluate the quality audio source separation. Within BSS_Eval, the *Source to Distortion Ratio* (SDR) measures the distortion between a target and the estimated multi-channel audio sources. With respect to high-frequency reconstruction, BSS_Eval gives more weight to lower frequency bands and penalizes more frequency content which is not in the target audio, even though this content might be perceptually relevant. In this sense, we recognise that a subjective listening experiment would be a critical important component of future work, but for this initial research, we adopt the SDR as our primary objective measure for this context. It is important to note that we exclude other metrics related to the artifacts, interference, and spatial distortion from BSS_Eval as these are designed particularly for source separation. The SDR is reported for each of the overlapping chunks of 30 seconds with a 15 second overlap.

#### 4.1.2. Time-frequency transform parameterisation

The STFT is computed using a Hann window of length $1024$ samples, which at a sampling rate of $44.1$ kHz corresponds to 23.2 milliseconds (ms), and a hop size of $512$ samples ($11.6$ ms).

The CQT is computed with the MATLAB toolbox in [18] using the default parameterization, with a minimum frequency of $27.5$ Hz, and a frequency resolution of 48 bins per octave. Up to the Nyquist rate of $22.05$ kHz this gives $463$ logarithmically-spaced frequency bins. Perfect reconstruction via the inverse CQT comes at the expense of high redundancy in time and results in $647$ time frames per second, *i.e.*, a temporal resolution of $1.5$ ms which is much finer than that of the STFT, while retaining a similar number of frequency bins ($463$ compared to $513$).

Since our goal is to reconstruct the higher frequency end of the magnitude spectrograms, we must contend with the fact that signal energy typically is much lower at higher frequencies than at the lower end. In the context of our convolutional neural network approach this creates a difficulty, since the high frequency magnitude spectrum we seek to predict may have very small values. To partially circumvent this issue, we can apply a logarithmic scaling to both the STFT and CQT magnitude spectrograms prior to training (and subsequently revert back to linear magnitude scaling prior to the eventual output signal reconstruction). However, before applying such a logarithmic scaling we must ensure all magnitude spectrum values (for both the STFT and CQT) are greater than 1, since any values below 1 will be negative after taking the logarithm, and thus ignored by the ReLU. To this end we apply the logarithmic scaling as follows: $\mathbf{X}_{\log} = \log_{10}(\alpha + \beta\mathbf{X})$, where $\mathbf{X}$ refers to either the STFT or CQT. For the CQT we set $\alpha = 1$ and $\beta = 4$, where as for the STFT no scaling is required thus we set $\alpha = 1$ and $\beta = 1$. The final stage of the pre-processing relates to deep learning methods usually requiring data to be normalized to an interval or include a batch-normalization step. Thus, we normalize all the training data to be between 0 and 1 by multiplying with a scale factor, which we set as the maximum of the training data.

To create the band-limited, *i.e.*, low-pass filtered versions of the music pieces for training (and subsequent reconstruction), we use an 8th order Butterworth filter. In order to explore two different conditions, we create one low-pass filtered version with a cutoff of $3500$ Hz and another at $7500$ Hz (approximately $f_s/12$ and $f_s/6$). For both, we seek to reconstruct the full remaining frequency range of the original recordings up to the Nyquist rate of $22.05$ kHz).

We split the STFT or CQT into overlapping chunks of $T = 30$ time frames with an overlap of $O = 10$. Chunks are randomly grouped each epoch into batches of $32$. For a fair comparison between bottleneck and stride-2 we use $N = 175$ of filters for bottleneck and $N = 40$ filters for stride-2, such that the number of parameters is equal for both of the architectures (1.8 million). The STFT is trained for 100 epochs. Since CQT has a higher time resolution, we generate more training data and we solely train the network for 32 epochs. The initial learning rate is $0.001$ for STFT and $0.0001$ for CQT.

#### 4.1.3. Implementation details

The code used in this paper is built on top of Pytorch, a framework for neural networks [3]. We ran the experiments on an Ubuntu 16.04 PC with GeForce GTX TITAN X GPU, Intel Core i7-5820K 3.3GHz 6-Core Processor, X99 gaming 5 x99 ATX DDR44 motherboard. Training a condition took 16 hours for the STFT and 44 hours for the CQT; by contrast, the enhancement stage runs faster than real-time on the same hardware. To ensure reproducibility, a fixed seed controls the pseudo-random number generation in Python. This is used when initialize the parameters of the CNN and to randomly split the dataset into training and testing. The results presented in Section 4.2 are for seed 0.

### 4.2. Results

The results for the bottleneck and stride-2 are shown in terms of SDR in Figure 5a and 5b for the CQT and STFT respectively. In each figure we present the SDR across the cutoff frequencies of $3500$ Hz and $7500$ Hz and show the difference in performance for examples in the training set versus those withheld for testing. Since we want to measure how much the quality of the reconstruction improves with respect to the low-pass input, we include the SDR for all the low-pass versions of the pieces in the dataset.

On inspection of the figures we can see that the best overall performance for the test set is obtained using the stride-2 architecture for the cutoff of $3500$ Hz and the bottleneck architecture for the cutoff of $7500$ Hz. In both of these conditions there is a negligible difference between the SDR on those musical recordings used for training, compared to those withheld for testing. In addition to the highest overall mean SDR values, we can additionally observe the greatest relative difference over the mean SDR of the low-pass filtered versions. For both approaches there is a relative increase in SDR of over $4$ dB. Since the SDR calculation is made directly on the waveforms, this suggests that relevant high frequency information from the original recordings is being reconstructed based soley on observing the band-limited versions.

When looking across the two architectures for the CQT results, we can observe that the stride-2 approach is less effective for the higher cutoff of $7500$ Hz. This may be due to the lower proportion of harmonic content above this cutoff, and hence the reduced impact of the stride's ability to model harmonic relationships.

[3] `http://pytorch.org`

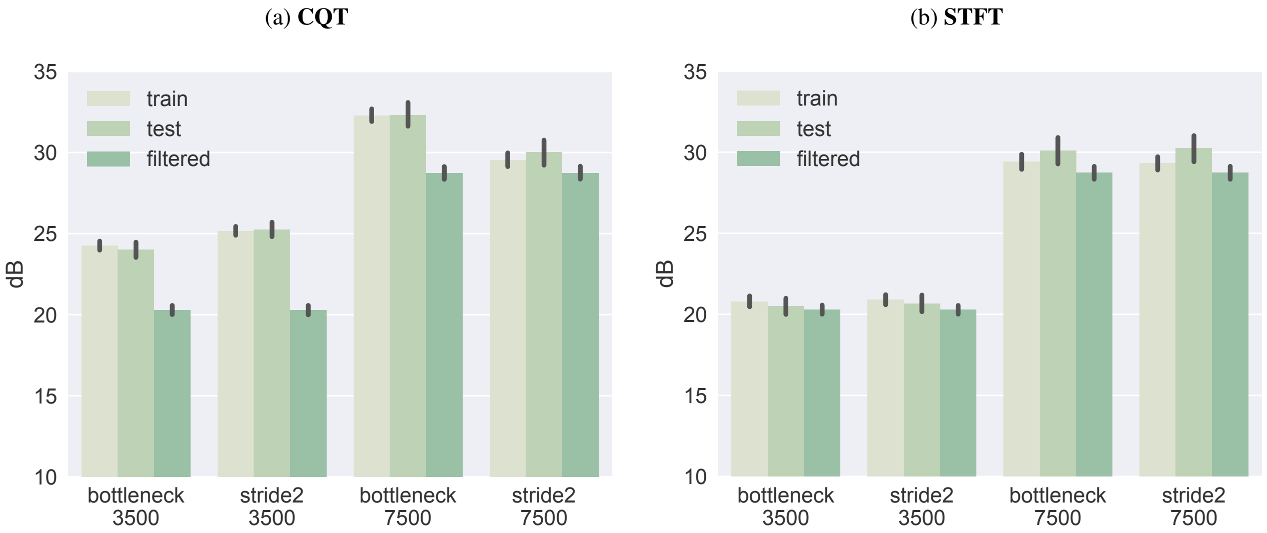

Figure 5: *SDR for (a) CQT and (b) STFT representations. The results compare the difference in SDR for training and testing sets, and the low-pass filtered condition (without enhancement), for the bottleneck and stride-2 CNN architectures and the cutoff frequencies of* 3500 *Hz and* 7500 *Hz. The black vertical lines represent the 95% confidence intervals.*

Looking at the comparison between the CQT and STFT, we can identify two main differences. First, the absolute SDR for the enhanced versions are lower than for the CQT across all conditions, and in turn, the relative improvement over the low-pass filtered versions is also reduced. This behaviour is in line with our original hypothesis concerning the advantage of using the CQT, where, although the frequency range to reconstruct is the same for both time frequency representations, the number of missing rows of the CQT is far smaller than that of the STFT. This is also consistent with results from image completion, in which larger image patches are more difficult to recover than smaller ones [10]. Another important factor may be the difference in temporal resolution for the two time-frequency representations, which is greater by a factor of approximately 8 to 1 for the CQT compared to the STFT; that while both process overlapping chunks of $T = 30$ time frames, the reconstruction of the CQT is much more localised in time than the STFT. We intend to explore this effect in future work by increasing the frame overlap in the STFT to a comparable level to that of the CQT. However, any significant increase in the frequency resolution of the STFT, *e.g.*, by using a larger window size would drastically increase the size of the model to be trained, and thus negate the approximately equal number of frequency channels in the STFT and CQT in our current setup.

To complement these objective results, we provide a set of short sound examples covering the eight reconstruction conditions, together with the original and two low-pass filtered versions. Furthermore, for the two best performing conditions: CQT stride-2 3500 Hz and CQT bottleneck 7500 Hz we provide an informal comparison of different approaches for phase reconstruction. To this end, we include phase reconstruction using: i) the low-pass filtered version (our proposed method); and ii) using low-pass filtered version below the cutoff and random phase above it. All of the sound examples are available at the following website: `http://telecom.inesctec.pt/~mdavies/dafx18/`

## 5. DISCUSSION AND CONCLUSIONS

We presented a new deep learning method to reconstruct the high frequency content of music recordings. Our evaluation demonstrates that due to to the logarithmic spacing of frequencies, the CQT offers a better time-frequency representation for this problem than STFT in terms of SDR. It is important to stress that these are initial experiments are performed under highly controlled conditions. Due to the high computational cost of training (which took several days using powerful GPUs), we only explored two cutoff frequencies, and used the same type of low-pass filter throughout. On this basis, we do not have sufficient evidence about the generalisation capacities of our trained networks to function under more arbitrary filtering conditions. This is especially important when considering our long term goal of the restoration of old recordings, for which we cannot assume any specific filtering conditions. Furthermore, in this scenario no stereo version of the recording may exist, which would require additional modifications to our approach.

Another important constraint within this study was the treatment of the phase in the reconstruction. While we do not provide unobservable information (*e.g.*, the phase of the original, full-band signal), our approach for using the low-pass filtered version phase could almost certainly be improved via the use of phase reconstruction techniques [32]. Since these are typically applied for an STFT representation, we intend to explore the means for doing this directly for in the invertible CQT representation in future work. Furthermore, we recognise the potential of using other time-frequency representations – provided that there is a method to invert them, *e.g.*, using Wavenet as a vocoder [33]. Furthermore, generative adversarial networks have recently became popular in image recovery and super-resolution [10] and can synthesize more realistic time-frequency content, which may yield further improvements to the quality of the signal reconstruction.

With respect to the evaluation, we acknowledge that BSS_Eval has been primarily designed for audio source separation, and further perceptual experiments are needed to better understand the subjective performance of our proposed method. Furthermore, BSS_Eval metrics do not always correlate with the perceived quality of separation [34]. In contrast to magnitude spectrograms, reconstructed images can be evaluated more directly because the inherent structure in the pixels can be understood in terms of the geometric and textural properties of scenes and objects. However in our approach the images correspond to time-frequency representations which are non-trivial for non-experts to visually interpret, and require an additional transformation stage to be audible. Within our training stage, the loss function relates to the mean squared error between the original magnitude spectrogram and the reconstruction, however our objective evaluation measures the SDR of the reconstructed audio signals, which explicitly includes phase information. Thus, we also intend to explore alternative loss functions (perhaps by using phase information directly) and subsequently investigate their correlation with perceptual ratings of audio quality from trained listeners. As part of this comparison we we intend to incorporate existing approaches for bandwidth extension which have been shown to be effective for music signals sampled at 44.1 kHz.

## 6. ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their expertise and generous feedback which improved the quality of this paper.

Matthew E.P. Davies is supported by Portuguese National Funds through the FCT – Foundation for Science and Technology, I.P., under the project IF/01566/2015. The TITANX used for this research was donated by the NVIDIA Corporation. "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-00020" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

## 7. REFERENCES

[1] H. Yasukawa, "Signal restoration of broad band speech using nonlinear processing," in *European Signal Processing Conference*, 1996, pp. 1–4.

[2] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, "Speech enhancement via frequency bandwidth extension using line spectral frequencies," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, vol. 1, pp. 665–668.

[3] D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrowband speech using non-negative matrix factorization," in *Ninth European Conference on Speech Communication and Technology*, 2005, pp. 1505–1508.

[4] E. Larsen and R. M. Aarts, *Audio bandwidth extension: application of psychoacoustics, signal processing and loudspeaker design*, John Wiley & Sons, 2005.

[5] D. L. Sun and R. Mazumder, "Non-negative matrix completion for bandwidth extension: A convex optimization approach," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

[6] Martin Dietz, Lars Liljeryd, Kristofer Kjorling, and Oliver Kunz, "Spectral band replication, a novel approach in audio coding," in *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.

[7] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 873–881, 2007.

[8] K. Li, Z. Huang, Y. Xu, and C-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015, pp. 2578–2582.

[9] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," *arXiv preprint arXiv:1708.00853*, 2017.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.

[11] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6721–6729.

[12] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[13] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016, pp. 1–6.

[14] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2017, pp. 258–266.

[15] T. Jehan, *Creating music by listening*, Ph.D. thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2005.

[16] N. Perraudin, N. Holighaus, P. Majdak, and P. Balazs, "Inpainting of long audio segments with similarity graphs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018, *(In Press)*.

[17] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with non-stationary Gabor frames," *14th International Conference on Digital Audio Effects (DAFx-11)*, pp. 93–99, 2011.

[18] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[19] P. Smaragdis and B. Raj, "Example-driven bandwidth expansion," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2007, pp. 135–138.

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, 2014, pp. 184–199.

[21] P. Jax and P. Vary, "Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden Markov model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, vol. 1, pp. I–680–I–683.

[22] X. Mao, C. Shen, and Y-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810.

[23] M. Miron, J. Janer, and E. Gómez, "Generating data to train convolutional neural networks for classical music source separation," in *14th Sound and Music Computing Conference*, 2017, pp. 227–233.

[24] M. Miron, J. Janer, and E. Gómez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 55–62.

[25] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: from coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[27] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[30] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive MIR research," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 155–160.

[31] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[32] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from STFT magnitude," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[33] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer," *arXiv preprint arXiv:1704.03809*, 2017.

[34] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–205, 2011.

[35] Christian R Helmrich, Andreas Niedermeier, Sascha Disch, and Florin Ghido, "Spectral envelope reconstruction via igf for audio transform coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 389–393.