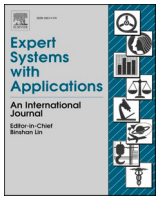
Expert Systems With Applications 230 (2023) 120586

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



A deep learning framework for audio restoration using Convolutional/Deconvolutional Deep Autoencoders

Alberto Nogales, Santiago Donaher, Álvaro García-Tejedor *

CEIEC, Universidad Francisco de Vitoria, Ctra. Pozuelo-Majadahonda km. 1, 800, 28223 Pozuelo de Alarcón, Madrid, Spain

ARTICLE INFO

Keywords:
Signal processing
Deep learning
Convolution
Autoencoders
Audio restoration

ABSTRACT

People communicate daily with their mobile phones and in some cases, the quality of the communication may be vital. Thus, there is a clear interest in improving the quality of communication in cases of low signal or interferences. This paper shows how deep learning techniques are used to restore audio files that simulate situations of background noise and loss of signal. Its main distinguishing feature is the direct use of the waveform instead of a spectrogram representation which lets the model be adapted to real-time communications or broadcasting. The results show that our proposal improves performance compared to Wave-U-Net. After restoring the audio, the difference between the original and the restored audio is, on average, less than 2%. In addition, a subjective test was carried out with 113 people who detected a significant improvement in the restored audio compared to the damaged one.

1. Introduction

Audio acquisition and reproduction are purely analog processes; the sound is captured by microphones that transform acoustic waves into electric ones. Speakers do the opposite: transforming electromagnetic waves into acoustics. However, the storage, handling/edition, and transmission can be either analog or digital.

The treatment of audio as an electrical signal falls within the field of signal processing which, according to Thon (2003), was first developed in the 1960s with the processing of sampled analog data by Franks and Witt (1960). Digital processing would come later, Godsill et al. (2002), with a sound signal converted into a stream of discrete binary numbers through an Analogue-to-Digital Converter (ADC). The reverse function (conversion of a digital signal to analog) is carried out by a Digital-to-Analogue Converter (DAC). The most common conversion method is Pulse-Code Modulation (PCM) where the amplitude of the analog signal is sampled regularly at uniform intervals through the sampling process. Digital audio depends on two main factors: sampling rate and bit depth. The first factor, the sampling rate or sampling frequency is defined as the number of samples per second that are extracted from the original audio, Pras and Guastavino (2010). The higher the sampling frequency, the more accurate the sound capture, and the digitized sound will be more faithful to the original (higher fidelity/quality). The second factor that determines digital sound's fidelity to the original analog signal is bit depth, which is the number of bits used to store each sample, Kipnis et al. (2015). A PCM stream must be encoded in a digital file with a given format for storage and manipulation. There are many different formats, but all require a great deal of information to reproduce sound. Several thousand values are required for a few seconds of audio. For example, one minute of a high-quality recording in WAV format requires 10 Megabytes of storage.

Even in the digital format, sound can be altered at different points of the end-to-end broadcast/receiver chain. Poor quality audio on the receiver side can be caused by any number of factors: poor transmission conditions, interference, signal loss, damaged files, etc. Audio restoration has attracted much interest from science and industry because voice calls remain one of the most common forms of communication and, on some occasions, the only one available. In sound transmission, much effort has been devoted to ensuring sound quality/fidelity even under non-ideal conditions given that, in audio communications, the better the audio quality, the better the receiver will understand the message. Low-quality transmissions may cause the information to be lost, distorted, or even entirely unintelligible at its destination. This is particularly important in an emergency when the clarity and intelligibility of transmissions can be a question of life and death. Solving these scenarios is the main motivation of the work by providing an audio denoising model that works directly with the waveform of the audio, avoiding the use of transformations, so it can be used in the most efficient way for

* Corresponding author.
E-mail addresses: alberto.nogales@ceiec.es (A. Nogales), a.gtejedor@ceiec.es (Á. García-Tejedor).

https://doi.org/10.1016/j.eswa.2023.120586
Received 21 March 2022; Received in revised form 10 March 2023; Accepted 27 May 2023
Available online 1 June 2023

real-time communications.

This paper explores the use of deep learning techniques to achieve audio restoration. Deep learning is a field of machine learning, defined by Bengio et al. (2015) as a technique that uses computational models composed of multiple processing layers (deep neural networks) to learn representations of data with multiple levels of abstraction. Our model is trained using input/output pairs of damaged and original audio and learns to perform two forms of audio restoration: audio cleaning (when noise appears on the main signal due to interferences of a secondary signal) and audio completion (when incomplete fragments are received due to loss of signal as in areas with poor mobile phone coverage). The model was evaluated in both objective and subjective ways: an objective test using some mathematical metrics and well-known baselines, speech-to-text machine applications and waveform analysis for both the original and restored audio files; and a subjective human survey, where individuals listen to different audio samples and rate them.

The work implements an improved version of the hourglass neural architectures adapted to the field of sound cleaning. The main innovation of our research is to work only with the waveform, without using any prior transform, such as the fast Fourier transform, the short time Fourier transform (STFT) or the frequency transform as in the works collected in the next section. This decision largely eliminates the introduction of artifacts and delays in the original signal to be cleaned. Working in waveform space means fast and efficient processing times, which are essential requirements for processing real-time streams such as telephone calls or radio and television broadcasts. It is in these environments where the use case analyzed in our paper occurs. Besides, working with spectrograms apply two transformations to the original signal: one from audio to spectrogram to train the model and the second from spectrogram to audio to recover the audible signal and be able to listen to it (as required in the examples of applications mentioned above). These two transformations imply, in turn, two problems intrinsically associated with this workflow. First, both transformations involve a loss of information that leads to worse audio quality. For this reason alone, our method would already provide better quality than other works, without even considering the improvements obtained by the proposed architecture. Second, the recovery of the audio signal from the spectrogram is not lossless. Some information (such as phase) is not included in the spectrogram and so, the obtained audios differ from the original. Most of the audios thus obtained are unintelligible or difficult to understand.

The most interesting contribution of our work, which benefits from working directly with the waveform, is the possibility of performing a subjective evaluation that in other works is either not done or is done in a very simple way and only with a few individuals. The reason for this is the great difficulty of returning an audio to its original state after performing a transformation such as the Fourier Transform since part of the original information is lost. Therefore, we have performed an exhaustive and extensive subjective evaluation in which a group of people listen to audios of different lengths to assess the quality of their restoration, evaluating the similarity of these audios with the originals. In addition, we have measured the percentage of information loss supported by our model by evaluating audios restored with a percentage of information loss varying between 10 and 90%.

Another contribution is the objective evaluation (which in our study has demonstrated the better performance of our model compared to well-known classical restoration methods like Wiener, Wave-U-Net, and DAP Design). The objective is complemented with two experiments. One performs speech-to-text to instances of restored and damaged audios and counts the number of words that match the original audio. The other compares the values of the waveforms in pairs of restored and original audios to obtain the percentage of different values. These results complement the subjective assessment, as the human ear can better pick up nuances and assess overall quality in real working environment circumstances.

This paper is structured as follows. Section 2 reviews some of the past work related to these issues. Section 3 describes the materials used in the research and the methodology applied to solve the problem. Section 4 presents the results of the various tests that were conducted. Finally, Section 5 offers some conclusions and possible areas for future work.

2. Related works

The history of audio restoration is a short one. The application of classic restoration techniques is reviewed in Czyżewski et al. (2012). In Esquef et al. (2001) high-frequency tones of an acoustic guitar are reconstructed using Commuted Waveguide Synthesis algorithm. Another model, using statistical techniques such as probability and estimation theory, is presented by Godsill et al. (2001). In Lu et al. (2003), audio restoration is achieved by synthesizing the missing parts based on waveforms while Cabras et al. (2010) use the Non-negative Matrix Factorisation technique. In Miura et al. (2011), audio restoration is done through recursive vector projection. In Sprechmann et al. (2013), the alignment of multiple copies of a recording is used to achieve the restoration. Mathai and Deepa (2015) achieve audio restoration in different cases by applying methods like Orthogonal Matching Pursuit (OMP) algorithm, spectral subtraction, or adaptative filters. Finally, Menendez-Ortiz et al (2018) use auditory masking properties for the frequency selection and the mapping to the intDCT domain.

Artificial Neural Networks (ANN) have recently been proposed as an alternative to classic techniques, specifically, Deep Learning methods that have proven efficient in the treatment of other types of signals and have been mostly applied to restore images and video. For example, Kawabe et al. (2019) use AlexNet to restore images of Braille texts. In Liu and Lam (2018), Convolutional Neural Networks (CNN) are applied to Poisson denoising in images. Ali et al. (2021) removes artifacts in endoscopy videos by applying Deep Learning models to detect them. CNNs are used by Zhang et al. (2017) for image denoising. Ulyanov et al. (2018), apart from image denoising solved problems like super-resolution or in-paining. These models are also used for reducing blur and noise in degraded images, De Vylder et al. (2016). Other models such as Autoencoders are used for hazy images, those affected by smoke, dust, or other particles, Yeh et al. (2018). In Dong et al. (2014), Super-Resolution Convolutional models are used to enhance images. There are also papers on the use of Natural Language Processing (NLP), by Salloum et al. (2017) and Náplava et al. (2018). The former uses Recurrent Neural Networks (RNNs) to restore missing punctuation in medical reports while the latter also uses RNNs but in this case for the diacritic restoration of Slovak texts. They have also been used to analyze electroencephalogram data, Schirrmeister et al. (2017) or radio signals, O'Shea et al. (2018). In Ashraf et al. (2021) Generative Adversarial Networks (GANs) are used to remove underwater ambient noise. A hybrid model using RNNs, and a genetic algorithm is used for signal restoration by Khan and Khan (2021).

The present work uses a Deep Learning model based on hourglass architectures, specifically U-Net, Ronneberger et al. (2015). In our research, the 2D layers initially used in U-Net for image processing will be substituted by 1D convolutional/deconvolutional layers for audio restoration. In Stoller et al. (2018), Wave-U-Net (an adaptation of U-Net) has been employed to separate singing voices in audio recordings which is a different use case to the one we want to solve. The paper focuses on the reconstruction of medium and high frequencies from damaged audio containing only low frequencies; that is, part of the audio is always available while our work restores all audio frequencies, even in cases where there is no information at all, or parts of the signal are damaged by noise. Although both use the same model, their architectures differ in the number of layers, neurons, and other hyperparameters. Also, Pandey and Wang (2019) use this model combined to do speech enhancement with an STFT for validation. The main difference with our work is the transformation to the frequency domain by applying an STFT during the process. In a third paper, Lu et al. (2013) work with Deep Autoencoders, trained with a linear search-based quasi-Newton optimization algorithm

and Mel frequency power (MFP) spectrum, which is used separately in each of the 3 hidden layers. If we compare it with our paper, we use a simpler model and do not need to transform audio into an MFP spectrum. Also, in Pascual et al. (2017), speech enhancement is approached using a U-Net architecture but in this case, this is the generator of a Generative Adversarial Network (GAN) which means applying a different model than in our case. Another work is Hsieh et al. (2020) which proposes a convolutional-deconvolutional model that integrates CNN layers with Stacked Recurrent Units for speech enhancement by applying an STFT in the audios. Again, a transformation into spectrograms is done, a task that is avoided in our work. Finally, Deng et al. (2020) use an LSTM recurrent neural network to improve the quality of audio based on its bitrates. Compared with our work not only a different model is used, but the application case is totally different.

Another distinguishing feature lies in the subjective evaluation we made by listening to restored audio. In most cases, this evaluation has not even been done. In Pascual et al. (2017), the evaluation consists only of one task evaluated by 16 listeners which is very little compared with our subjective evaluation. Although the subjective evaluation does not seem important, it should be considered that in the works that use spectrograms, this evaluation might not have been done as converting spectrograms to audio could lead to loss of audio quality so they could not be evaluated by listening to them. The present work uses a Deep Learning model based on U-Net, as a paradigm of hourglass architectures, but applied to 1D signal (sound) instead of 2D images, as it was originally designed. The benefit comes from the idea of using U-Net-like networks for a different domain under the assumption that it will behave similarly, as demonstrated in our work.

3. Materials and methods

This section describes the datasets used to train the model with a theoretical description of the models used to solve the problem.

3.1. The initial dataset

It is well known that neuronal network models are only as powerful as their training data; thus, the adequate selection of the dataset is paramount, and, in this case, it means selecting the correct audio files as input data. As mentioned above, there are many different ways to store digital audio. The choice of the file format, audio quality, and content is important as this will determine the success of the research project.

The chosen format was WAV (apocope of WAVE Audio File Format), developed by Microsoft in the early 1990s from its RIFF (Resource Interchange File Format) specifications for multimedia file storage. The choice of this format was made based on Siegert et al. (2016), firstly because it is an uncompressed format (that is, what is stored on the computer is what the user hears) and secondly, it has been a standard for almost 30 years. It also supports a wide range of quality formats allowing multiple sampling rates and bit depths.

The second decision was related to the sampling rate and bit depth, the most important features of audio quality Kanetada et al. (2013). The number of channels was also decided, that is, the number of different tracks the file contains. It was decided to use a sampling rate of 16 kHz, with a 16-bit bit depth and a single channel considering the computer hardware (CPU Intel(R) Core (TM) i5-6500 @ 3.20 GHz, 32 GB of RAM, GPU Nvidia GeForce GTX 1080Ti with 11 GB GDDR5X) and to work with good quality audio that could be processed in a reasonable time.

For our case, the best training datasets are composed of conversations between people, that is a human-speech dataset, due to the similarity of these audio files with calls with interference or signal loss. A search was made for datasets with damaged audio but very few were found, and often of dubious integrity and poorly documented. It was decided to create a set of training data using good-quality audio files and subsequently transformed it to build the two use cases.

The TED-LIUM 3 dataset, Hernandez et al. (2018), was found to be suitable. It was created from the audio recordings of the famous TED-TALKS series, also containing audio transcriptions, given their use in Speech-to-Text applications. There are 2,351 audio talks in NIST Sphere format (SPH), a common format for audio speech files containing both audio and text content. There was a total of 452 h of audio. The dataset consisted of 54 GB of compressed data (175 GB after decompressing), with a sampling rate of 16 kHz and a bit depth of 16-bit. Each file contained only one channel.

The set consisted of English language audio content consisting of different people with different tones of voice on different types of recordings under different circumstances. These differences bring more value to the work. The dataset was downloaded from Open Speech and Language resources.[1]

3.2. Data transformation and pre-processing stage

The first transformation applied to the original audio dataset was to change the file format. Files were converted from the original SPH to WAV format using a command-line script for the SoX[2] tool used to convert audio files.

Once the files were encoded in WAV format, the actual pre-processing could begin. A Python script was built to transform the WAV files into their numerical representation. WAV files were read using the SciPy[1, 2, 3, 4 and 10] library, an open-source module for science, mathematics, and engineering. The files were transformed into arrays using NumPy,[4] the basic Python library for scientific computing. Finally, all the data was stored in a CSV file using pandas,[5] a library for managing data structures.

For a given sample rate and bit depth, the amount of data to be processed depends on the length of the audio fragment. The longer the fragment, the greater the amount of data. The files vary in size depending on the varying duration of the talks used in the original data set. It was necessary to define in advance the length of the fragments fed to the input layer of the neural model as this determines the size (number of neurons) of the input and output layers and the computational power required. Finally, it was decided to work with two-second audio recordings, long enough to understand the speech while maintaining a manageable size, with arrays of size 32,000 size. At this point, rows in the CSV file were divided into fragments of this length, obtaining 10,880 instances for the training stage.

It was also seen that the numerical representation of the audios' ranges varied widely: real values between −1.0 and 1.0 or integer values in the range of −30,000 to 30,000, due to the different recording conditions at the source. This makes it necessary to normalize the NumPy arrays. Normalization is a crucial process during pre-processing. According to Sola and Sevilla (1997), it reduces the estimation of errors, speeds up the training, and avoids the neural network to give more importance to some files over others. The generalized form of the min–max feature scaling for the interval (-1.0, +1.0) was used so in each file the sample values range from 1.0 or −1.0. These were calculated using Eq. (1):

$$x_n = 2\frac{x - minval}{maxval - minval} - 1 \tag{1}$$

where x is the original value, x_n is the normalized value and *maxval*, and *minval* is the maximum and minimum file values, respectively.

Another important factor in data normalization is the choice of the number of decimals to be used. We decided this by conducting a human survey where 10 candidates listened to four different samples of

[1] https://www.openslr.org/51/.
[2] https://sox.sourceforge.net/.
[3] https://docs.scipy.org.
[4] https://www.numpy.org/.
[5] https://pandas.pydata.org/.

randomly selected five-second audio fragments. The samples were built using 8, 4, 3, and 2 decimals of precision. The aim was to determine at what resolution users could perceive a qualitative difference. Additionally, one of the four samples was randomly repeated. The candidates listened to each recording as many times as wanted and gave a numerical score: 1 bad quality, 2 average quality, and 3 good quality. The survey showed that 3 decimals of precision would be appropriate, as from that value upwards none of the survey participants (0/10) could notice any difference in quality whereas with 2 decimals of precision some participants perceived lower quality.

3.3. Building the training data

Deep Learning models, like all neural networks, require a training phase to adjust their weight matrix to solve the proposed problem. For supervised models, as in this case, a training set consisting of pairs of input/output data is required. Therefore, pairs of original sample audio files with their corresponding damaged versions (noisy or incomplete) were created, the input being the damaged version and the output being the original audio stream. The procedures used to process the audio fragments are described below.

Audio with noise. Noise is a broad concept that can include several alterations to the original signal. Therefore, it is very difficult to reproduce actual noise conditions in an audio wave. A common case is an interference, noise due to the intersection of two waves, for example, two radio waves. When this occurs, parts of the original audio values are replaced by waves that cause interference and generate noise. Therefore, the training data set was constructed by randomly replacing 40% of the samples in the original audio files with values obtained from a Gaussian White Noise sinusoidal wave (GWN), a technique inspired by Othman (2016) and Balakrishnan and Mazumdar. This wave was composed of random values between −1.0 and 1.0 under a normal distribution with an average of 0. Fig. 1 shows an example of the original audio (above) and the same audio with noise (below). The damaged version has a combination of original samples and new samples generated by the GWN. The shape of both waves shows partial matches.

Incomplete audio. This occurs when part of the information is lost in communication for some reason, and incomplete information (partial signal loss) reaches the receiver. In this case, some of the samples in the audio fragment have no value. To simulate this type of damage, a percentage of 70% random samples from the original audio was set to 0, leaving a resulting wave set of some original values and randomly positioned 0 s.

Fig. 2 shows an example of the original audio (above) and the same audio damaged by a loss of information (below). The difference is that the damaged audio retains some of the original audio but most of the samples are 0. As shown, the shape of the wave is similar but some of the samples have a value of 0.

These two techniques were applied to the entire dataset of original audio files to satisfy the two use cases. A DataFrame was built containing in each row the numerical representation of a different original audio fragment and its two use cases: audio with noise and audio with signal loss.

This DataFrame has a shape of 10,880 × 3 (rows correspond to audio instances and columns the original audio plus the two proposed use cases). Each of these cells contains audio represented as an array of 32,000 values. The final version of the DataFrame was exported as a single CSV file[6] containing all the data needed for the research. This file occupies 8.6 GB in total.

3.4. Convolutional-deconvolutional neural network

To solve the two use cases presented in this paper, a neural model based on U-Net was developed and trained using two different data sets for each of the two use cases. The architectural decisions that led to the present model aimed to improve the results of previous attempts based on solid and already-proven concepts. These concepts are described in the following.

Autoencoders. In Hinton and Salakhutdinov (2006), an Autoencoder is defined as "an adaptive multilayer encoder network to transform the high-dimensional data into a low-dimensional code and a similar decoder network to recover the data from the code". This definition describes the architecture used in this work. First, a small representation of the input features was obtained by going through the different layers of an encoder. Then, a decoder raised the feature representation by obtaining an output of the same dimension as the input data.

U-Net. Given the good results in image segmentation using U-Net, it was decided to build another architecture of similar inspiration. Karimov et al. (2019) define U-Net as a convolutional Autoencoder that can also receive the name "hourglass model" due to its shape. In this case, the greatest contribution of U-Net was to add deconvolutional layers and skip connections to an Autoencoder. Skip connections allow the problems produced by the bottleneck of the architecture to be addressed. The bottleneck is found between the encoder and the decoder where feature representation is obtained. At this point, data is so reduced that some important information could be lost and not transmitted to the decoder. To solve the problem, skip connections can be established between the encoding and decoding layers.

Convolution and Deconvolution in 1 Dimension. 1D Convolutional Neural Networks (1D CNNs) were first presented by Kiranyaz et al. (2015) as an adaptation of the classical 2D CNNs for classifying 1D signals like electrocardiograms (ECGs). 2D CNNs, as now understood in Deep Learning, were first introduced by LeCun et al. (1999). These can extract particular features in a delocalized way which made it possible to obtain results never before seen in the classification of images, Krizhevsky et al. (2017). By using these models, a feature found in part of an image can then be found in another part of another image. In this case, the main features of the audio will be found.

3.5. The proposed solution

At this point, in line with the above, a 1D Convolutional/Deconvolutional Deep Autoencoder based on U-Net was developed. The decision of using 1D Convolutional models improves the method in terms of being used for real-time communications as no transformation from raw audio to image representation has to be done, so we can work directly with the waveform of the audios.

To train this model, one of the damaged audio files is used as input. Its size is reduced, features extracted, and then increased to its original size, and the model creates audio of the same size as the output. This audio is compared to the audio in good quality, and with that comparison, the model learns where it has made a mistake, adjusts its parameters, and learns. In this way, after training using the whole dataset, the model has learned how to restore a damaged audio file. The training consists of two steps: first, we train a model with the noisy audios until it performs well, and then, we train this model with the dataset of audios with low signal.

The model has an input layer that connects the 32,000 values of the array (audio fragment with only one channel) with the next layer. The input layer propagates this information to the first convolutional block. This is where the dimensionality reduction or convolution stage begins. This stage consists of four convolution blocks, where each block produces a feature map of input audio. Each of these blocks is composed of two 1-dimension convolution layers followed by a pooling layer of size 2. Convolution layers of the same block have the same number of neurons, and all of them use the Rectified Linear Unit (ReLU) function as an activation function. ReLU returns the highest value between 0 and the input value. Filters in the convolutional layers always have sizes 3x1 and stride 1. Fig. 3 shows the architecture used in this work.

[6] https://zenodo.org/record/6109421#.YpkGP5NBw6A.

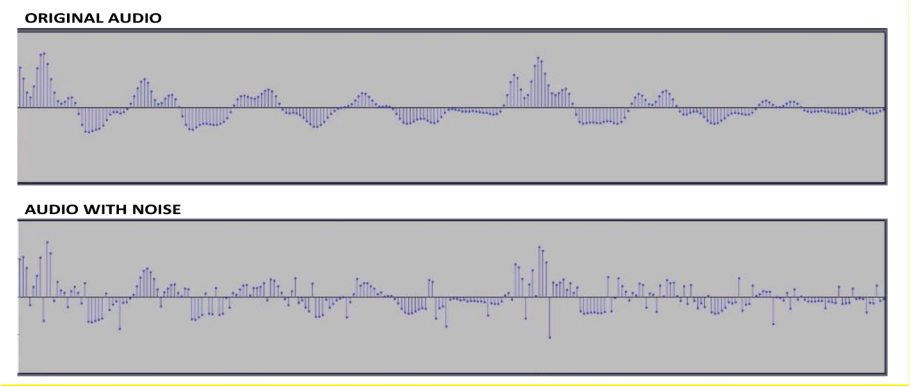


Fig. 1. Comparison between the waveforms of original complete audio and audio damaged by noise (they differ in 40% of the values).

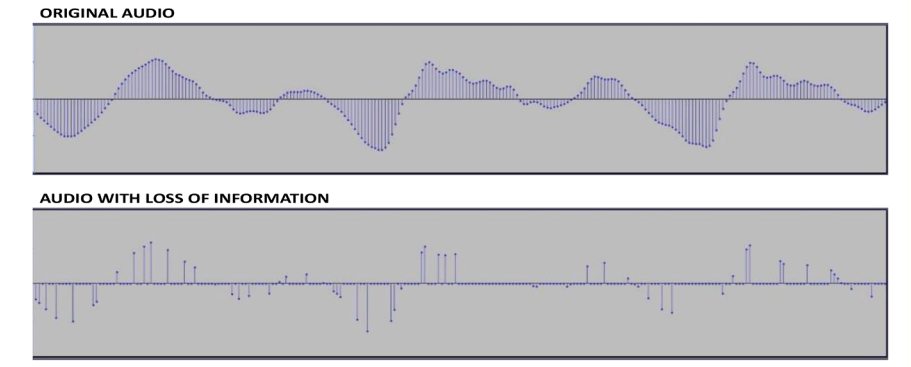


Fig. 2. Comparison between the waveforms of original complete audio and audio with loss of information (they differ in 70% of the values).

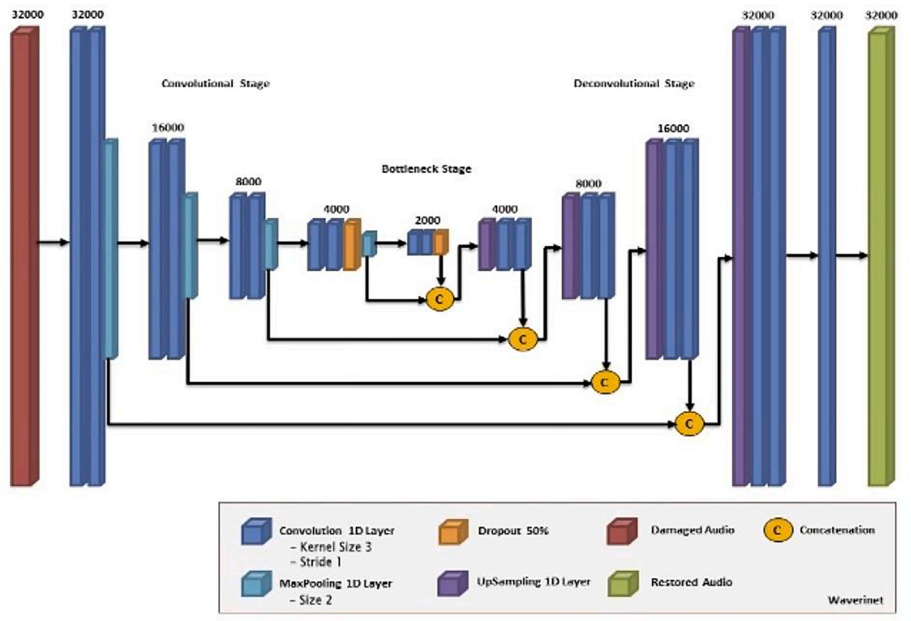


Fig 3. Description of the architecture used in the work. The convolutional stage corresponds to the encoder and the deconvolutional stage to the decoder.

The number of neurons between each block increases, starting with 64 neurons in the first block, 128 in the second, 256 in the third, and 512 in the fourth. This reduces the size of the input audio feature map more and more. At the end of this reduction process, it has gone from an array of 32,000 values to 2,000, with all essential features of the input audio being extracted and stored, that is, the essence of that audio has been found.

At this point, the bottleneck stage begins. This stage consists of a single convolution block, but without the final layer of pooling. It is composed of two convolution layers with 1,024 neurons each. During this stage, the feature map remains at 2,000 sizes.

The stage of increasing dimensionality, or deconvolution now

begins. This stage has four deconvolution blocks, the same number as convolution. Each of the blocks is composed of an upsampling layer followed by two convolution layers. Again, the number of neurons within the layers of the same block is the same, but between each block it is different and all of them again use the ReLU activation function. The number of neurons per layer of each block is 512 for the first, 256 for the second, 128 for the third, and 64 for the fourth. This increases the dimensionality of the feature map. In this process of deconvolution, from the very small feature map in the bottleneck stage, new audio is created using only the essence of the original audio and the extracted features. Lastly, a final convolution layer with two neurons is used to achieve the same final audio size as the input. The last layer, the output layer, uses the hyperbolic tangent (tanh) activation function so that values in the range −1.0 to 1.0 are obtained from it, the same range of values that were introduced in the input layer.

Furthermore, it was necessary to make use of other techniques for this reconstruction to be the closest possible to the original audio. Firstly, in the last convolution block and the bottleneck stage, it was necessary to use dropout, a technique that makes some connections between neurons randomly disconnected during the training process to force information to flow through each of the connections. In this way, the whole network learns, and no neurons or connections remain under-trained or over-trained, known as under or overfitting. It was necessary to implement a 50% dropout between these two layers. Thus, in each training stage, 50% of the connections were disconnected.

The other technique necessary was the use of skip connections, that is, the connection between non-consecutive layers. These jumps in the connections were established between the end of the convolution and deconvolution blocks with the same number of neurons, that is, with their symmetrical version in the other stage of the model. This means that the output that produces a convolution block is, at the same time, the input for the next convolution block and partial input of the symmetrical deconvolution block, which also receives a feature map of the deconvolution block before it. The key is to make use of concatenation layers where two inputs come together in a single feature map, which is the input for the deconvolution block. This mechanism maintains some features of the audio that were lost during the whole process.

Despite using U-Net as inspiration, most of the hyperparameters chosen in this work were tuned for the proposed use cases. Convolutional layers in U-Net were 2-dimension designed for image processing with input data being matrixes 572x572 and sigmoid loss function at the output layer. The proposed architecture has one-dimension convolutional layers adapted for arrays of size 32,000 as input data and hyperbolic tangent as loss function at the output layer.

Implementation of the model was carried out using Keras,[7] an API developed for Python that runs over TensorFlow and facilitates the creation and development of neural networks. Thanks to this API, it was possible to declare the shape of the tensors (data structures with which the neuronal networks work) and build the neuronal network itself layer by layer.

3.6. Training phase

The training dataset split using 80% for training/validation and 20% for the test, comprises a total of 10,880 arrays representing 2-second audio fragments without overlapping. Two twin models were trained, each one for a different use case, with the following training sets:

- Case 1. Input: Audio with signal loss. Expected Output: Original audio (without loss signal).
- Case 2. Input: Audio with noise. Expected Output: Original audio (without noise).

In both cases, the parameters used during the training were:

- Learning Rate: 0.001. This parameter measures the size of the steps when training the model to reach a local minimum, Ruder (2016).
- Optimizer: Adam, an adaptive learning rate optimization algorithm designed specifically for training Deep Neural Networks, Kingma and Ba (2014).
- Loss: Mean Squared Error (MSE) is the way error is measured when training the model. MSE calculates the Euclidean distance between denoising audio and its ground truth (the audio without noise), Venkataramani et al. (2018).

The training stage consisted of tuning 10,812,677 parameters in an Intel(R) Core(TM) i7-8700 K CPU @ 3.70 GHz with 32 RAM Gigabytes and GeForce GTX 1080 Ti GPU that lasted 13 min per epoch. If we compare it with other works in the literature, they have more than 56 million parameters in Pascual et al. (2017) and 6.4 million in Pandey and Wang (2019).

After training the MSE was reduced to 0.0025, meaning that the prediction made by the network failed by only 0.05 in the working values range, between −1 and 1, i.e., it failed around 2% of the values of the predicted samples concerning the expected output samples.

4. Results and evaluation

This section presents the obtained results. Given that the data consists of reproducible audio files, what the network generates is the restored audio and the results will therefore be studied on these audio files for the different use cases. The evaluations are divided into objective and subjective, the results of which will allow us to reach firm conclusions.

4.1. Objective evaluations

These evaluations use software or other means that do not introduce biases as may occur in human evaluation.

Comparative baselines. We have evaluated the performance of our model against some baselines. Lim and Oppenheim (1978) present a classic method in the field called Wiener filters to reduce the noise in the frequency domain. Wave-U-Net is a neural model used for music and human speech segmentation, Stoller et al. (2018). It is based on a convolutional U-Net architecture, and it can separate the human voice from its background noise. Finally, a more recent proposal is from Narayanaswamy et al. (2021) which uses U-Net by adopting dilated convolutions with an exponentially increasing dilation schedule,

To evaluate the performance of the baselines, we have created a new dataset, different from the test set used during the training stage, executing our model. We refer to this dataset as evaluation one, it comprises 30 audios of 5 s, 30 of 10 s, and 40 of 30 s. As in the training set, we have added background noise and applied a filter for loss of information. The audios can be found in Spanish (which is not the language used in the training set) and English, belonging to different themes (cooking tv shows, political speeches, or medieval tv series) with their different recording methods. These audios do not belong to the initial dataset of TED talks used to create the training dataset. The metrics that evaluate the dataset have been Mean Absolute Error (MAE) and MSE. MAE measures the error between two instances of the same phenomenon. Qi et al. (2020). Performance results for information loss can be seen in Table 1 while Table 2 shows results for the noise use case.

Previous tables demonstrate that our model performs better than the selected baselines. The error is interpreted as follows; the lower it is, the better the reconstruction is done.

The information in the two tables above can be summarized in Fig. 4. We have used a diagram bar, so we can observe how the error affects the different models in proportion. As can be seen, our method is by far the one that makes the slightest error. Only in the case of audios of 10 s for

[7] https://keras.io/.

Table 1
Evaluation of the present model against Wave-U-Net for audios with loss of information using the MSE and MAE comparing three different lengths of audio: 5, 10, and 30 s.

Audio length		MSE	MAE
5 s	Wiener	0.0161	0.0703
	Wave-U-Net	0.0136	0.0625
	DAP Design	0.0338	0.0992
	Our model	**0.0010**	**0.0172**
10 s	Wiener	0.0156	0.0690
	Wave-U-Net	0.0131	0.0609
	DAP Design	0.0316	0.0949
	Our model	**0.0101**	**0.0164**
30 s	Wiener	0.0147	0.0668
	Wave-U-Net	0.0116	0.0571
	DAP Design	0.0264	0.0825
	Our model	**0.0008**	**0.0147**

Table 2
Evaluation of the present model against Wave-U-Net for audios with noise using the MSE and MAE comparing three different lengths of audio: 5, 10, and 30 s.

Audio length		MSE	MAE
5 s	Wiener	0.0113	0.0608
	Wave-U-Net	0.0098	0.0599
	DAP Design	0.0338	0.0992
	Our model	**0.0047**	**0.0484**
10 s	Wiener	0.0112	0.0603
	Wave-U-Net	0.0096	0.0592
	DAP Design	0.0316	0.0949
	Our model	**0.0044**	**0.0466**
30 s	Wiener	0.0108	0.0593
	Wave-U-Net	0.0090	0.0599
	DAP Design	0.0264	0.0825
	Our model	**0.0037**	**0.0425**

loss of information, it can be compared with Wave-U-Net. We can also see that DAP Design is the worst at solving the presented use cases. Finally, the graphic shows that there are no big differences in the baselines' performance comparing the two use cases. In the case of our method, it works much better for the loss of information use case.

Speech-to-Text test. Speech-to-Text "is the ability of machine/program to identify words and phrases in spoken language and convert them into the machine-readable format", Trivedi et al. (2018). The tool used in this case is Sonix,[8] an audio and video transcription tool that uses artificial intelligence. The experiment consisted of three stages. First, the original audio was transcribed, and the number of words was counted. Then, the corresponding damaged audio was then also transcribed and the number of coincidences with the original was counted. Finally, the audio restored by the neural model was also transcribed and the coincidences with the original audio were counted again. The experiment was conducted with a new evaluation dataset formed by 30 audios of 5 s, 30 of 10 s, and 40 of 30 s. Table 3 shows the average results with its standard deviations for both use cases (noise and loss of information) grouped by audio duration.

As shown in Table 3 in all cases our model improves the results. Improvements in the case of information loss are around 80% for audios of 5 and 10 s, due to the bad performance of Sonix when these audios are degraded. In the case of audio with noise, the improvement is always around 40%. Sonix, applied to our evaluation dataset of restored audios, can identify at best only 85.55% of the words, and 78.53% in the worst case. These results should be put in context after measuring the performance of Sonix manually by counting the number of coincidences between the original audio transcribed by the tool and a manual transcription made by us. With this experiment, the performance of Sonix is 89.7%.

In Fig. 5, we compile the information from the previous Table. It uses pie charts, so we can the percentage of matching words and how it increases from damaged audio to restored ones. The graphics show that for audios with loss of information, the number of matching words augments a lot except for audios of 30 s in which the amount is only tripled. In the case of audio with noise, the percentage of matching words augments by more than 25% for the different audio lengths. If we compared the use cases between them, we could see that in the damaged audio with loss of information is more difficult to detect words but when they are restored the number of detected words have no significant differences between use cases. So, we can conclude that adding noise to the audio damages them less.

Waveform comparison test. A second way to objectively prove the results is through visual perception. The audio files can be represented in waveforms (diagrams that represent the values of the samples versus time) and it is possible to see the differences between the different audio files.

Firstly, the waveforms for the different use cases were obtained using Audacity,[9] a free, open-source, cross-platform audio software. For each of the use cases, the damaged audio, the restored audio, and the original audio were plotted. Thus, at first sight, the restored and the original audio are almost equal but very different from the damaged one. This is shown in Appendix A, Figs. 1 and 2. As this test cannot be understood as 100% valid, a more accurate test was conducted using waveforms.

The similarity between the restored and the original waveform can be measured using FourierRocks,[10] a WAV file comparison tool. The experiment consists, again, of taking 100 random audio WAV files with durations of 5, 10, and 30 s (30, 30, and 40 instances respectively). The comparison is made in the time domain, which means that it measures how the signal changes over time. For each comparison, a percentage of error is given. This percentage means the number of different values by comparing both waveforms on a value-by-value basis. Results are provided in Table 4.

As shown in the previous table, the percentage of error in the 100 tests is very low, from 0.65% to 0.83% in the case of noise and from 1.37% to 1.62% in the loss of information case. The average error value is 0.73% for the first use case and 1.49%, meaning that the original and restored audio files have minimal differences. It can be concluded that the audio files are very well restored in both cases with better performance in the audio files with noise.

A bubble chart has been provided in Fig. 6 to show which combination of use case and audio length has a greater number of equal values when comparing their waveforms. As can be seen, the use case of audio with noise works better in this case. This can be related to the fact pointed out in the previous evaluation test, that the audios with noise are easier to listen to. We can also check that the length of the audio does not affect the restoration process.

4.2. Subjective tests

The performance of the model was also evaluated with human involvement. These tests are considered subjective since they depend on the person making the evaluation. The results provided in this section consist of statistics obtained through human surveys.

Listening tests. Several volunteers were surveyed through a Google form questionary to find out how they perceived the differences between the various audio files. A total of 113 people responded, aged from 18 to 73 years old, from college students to PhDs. It should also be noted that 12 of the volunteers declared some hearing condition. In the survey, damaged audio was played, then the same restored audio by the neuronal model, and then the corresponding original audio. After the

[8] https://sonix.ai/.

[9] https://www.audacityteam.org/.

[10] https://sourceforge.net/projects/fourierrocks/.

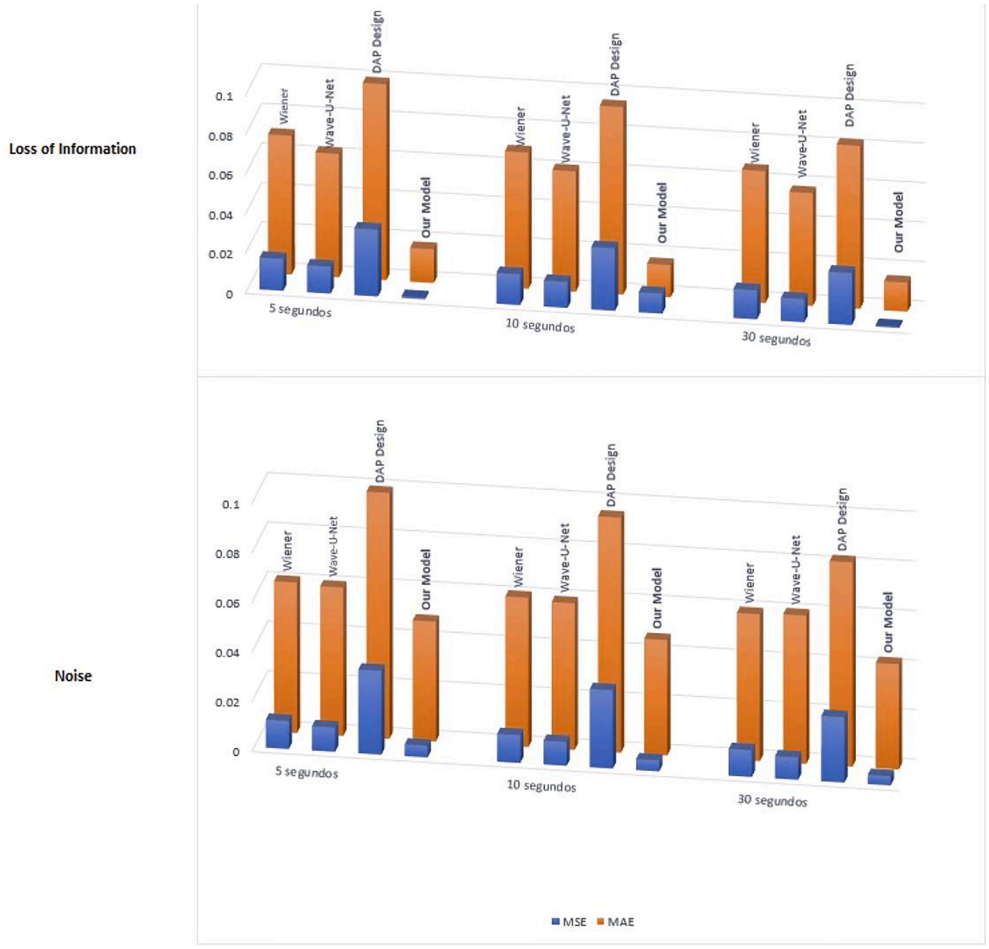


Fig. 4. Comparison of the errors provided by the proposed model with different baselines depending on the two use cases described by diagram bars.

Table 3
Evaluation of the two use cases by counting the number of matching words after using the audios with a speech-to-text tool.

Audio type		Matching words in damaged audio	Matching words in restored audio
5 s	Loss of information	2.74% (±0.09)	79.29% (±0.27)
	Noise	42.65% (±0.36)	80.62% (±0.28)
10 s	Loss of information	3.54% (±0.12)	78.53% (±0.17)
	Noise	48.02% (±0.28)	85.86% (±0.09)
30 s	Loss of information	21.04% (±0.21)	81.38% (±0.12)
	Noise	45.83% (±0.21)	85.55% (±0.09)

respondents listened to the three audio files as many times as they wanted, the repaired audio was given a numerical score, according to its quality compared to the damaged and the original audio. The score consisted of 4 possibilities depending on their perception: "Same as the damaged", "Some improvements over the damaged", "Much better than the damaged" and "Practically same as the original". The survey showed 15 different audio files (5 for each duration of 5, 10, and 30 s) for both use cases, the same set for all volunteers.

Noise. All respondents observed very significant improvements over the damaged audio, indicating in more than half of the cases that the restored audio was much better than the damaged one. But it should also be noted that in more than 25% of the cases almost all noise was removed from the damaged audio. A comparison of the results as a function of audio duration shows that the variations are virtually the same. Of the total responses, between 26.2 and 28.7% (depending on the three use cases of audio length) indicated that the repaired audio is practically the same as the original audio. 56.6 to 58.9% said that the repaired audio is much better than the damaged ones. Also, in the case of some improvements over the damaged one, it ranges from 14% to 15.2%. In less than 1% of the cases, the repaired audio was the same as the damaged one, as can be seen in Fig. 7.

Loss of information. In this case, all participants observed a considerable improvement in the audio files compared to the damaged files and in all audio files, an improvement was noticed to a greater or lesser degree. Of the total answers, 54.0 to 56.1% said that the repaired audio is much better than the damaged one. The perception that the repaired audio is practically the same as the original occurs in 14.9% of the respondents, 18.8% for the short audios, and exceptionally 24.1% for the audios of 30 s. These percentages are 26.9, 28.1, and 20.7 when audios had some improvements over the damaged one. In less than 1% of the cases, the repaired audio was judged to be the same as the damaged one. These results are shown in Fig. 8.

Several conclusions can be drawn from the simultaneous analysis of both figures. First of all, it is confirmed again that audios with noise are restored slightly better than audios with a loss of information. In the second case for audios of 5 and 10 s, the perception "Practically same as the original" drops to percentages of 18.8 and 14.9 respectively. The consequence is that in the first use case, the two best perceptions sum up to around 85% of the evaluations for the three audio durations while in the second case percentages are a bit over 70 for audios of 5 and 10 s. These latter cases should be studied in-depth.

4.3. Other results

Once proven that the system is capable of reconstructing damaged

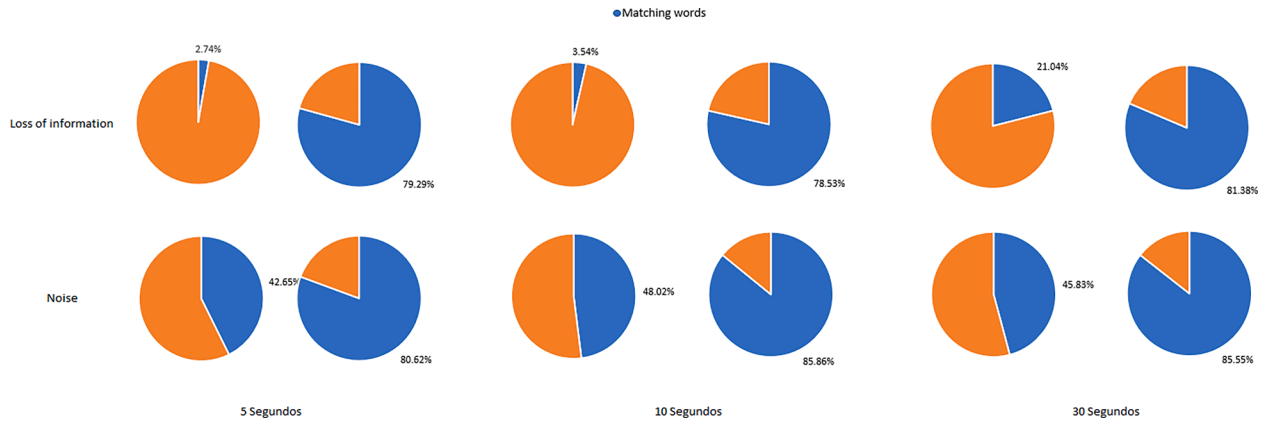


Fig. 5. Comparison of the errors provided by the proposed model with different baselines depending on the two use cases described by diagram bars.

Table 4
Evaluation of the two use cases with three different audio lengths comparing the percentage of equal values in the waveforms.

Sample type	Noise	Loss of information
5 s	0.68%(±0.14)	1.62%(±0.39)
10 s	0.65%(±0.10)	1.49%(±0.30)
30 s	0.83%(±0.30)	1.37%(±2.49)
Average	**0.73%(±0.23)**	**1.49(±2.49)**

audio files, a study was carried out to determine the reconstruction limit. In other words, what amount of noise can be eliminated (for audio files with noise), and what percentage of missing information can be recomposed (for audio files with loss of information). A study was conducted by rating the restored audio files. This rating ranges from 0 to 3. 0 means that no improvements were found compared to the damaged audio. 1, that the restored audio is somewhat better than the damaged file. 2 that the reconstruction was done with good quality, and 3 that the restored audio is as good as the original.

Table 5 shows the results for the case of noisy audio files. In this case, the percentage in the left column represents how much original information contains the audio. 90% means that 90% of the information is from the original audio, while the other 10% is produced by the GWN. In the evaluation, for scores 2 and 3, the content of the speech was able to be understood, while for 0 and 1 it was not. As shown in Table 6, except for 10% of all the restored audio files were understandable. We have also added the MSE and MAE to evaluate how the error evolves.

Table 6 shows the results in the case of loss of information. The left column shows the percentage of original information remaining in the damaged audio. When the value is 30%, it means that 70% of the original audio information is missing, and it only contains 30% of it. 90% means that it has only lost 10% of the original information. In cases with marks 2 and 3, the content of the speech was considered understandable, while for marks 0 and 1 it wasn't. As can be seen, audio can be reconstructed even with a loss of information up to 80% from the original audio.

5. Conclusions and future work

The goal of this research is the restoration of audio files damaged by noise or loss of information through the application of deep learning techniques. It was decided to use speech audio files rather than music or other content because these audio files are like the use cases of phone calls proposed at the beginning. In speech files, it is also easier to evaluate objective (word count) and subjective (perceived understandability) results. Files were found and prepared for the research. To best

Fig. 6. Comparison of use cases and audio lengths using a bubble chart, so the proportion between cases can be seen.

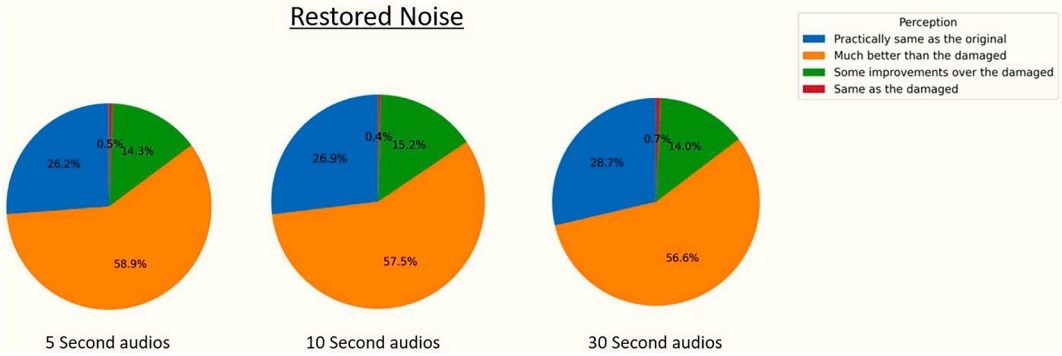


Fig. 7. Comparison of pie charts of audio files with different lengths for the noise use case to evaluate the subjective perception of people.

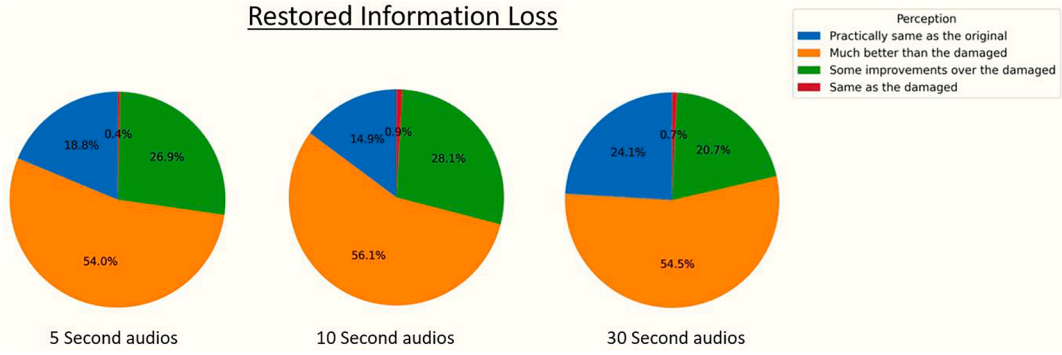


Fig. 8. Comparison of pie charts of audio files with different lengths for the loss of information use case to evaluate the subjective perception of people.

Table 5
Evaluation of audio files with noise changing the percentage of original values that must be restored using a subjective scale depending on the perception of people.

Percentage of audio with the original one (%)	Evaluation	MSE	MAE
90 %	3	0.00433	0.03930
80 %	3	0.00260	0.02859
70 %	3	0.00155	0.02169
60 %	3	0.00112	0.01784
50 %	3	0.00077	0.01487
40 %	2	0.00056	0.01296
30 %	2	0.00040	0.01146
20 %	2	0.00030	0.01048
10 %	1	0.00025	0.00999

Table 6
Evaluation of audio files with loss of information changing the percentage of original values that must be restored using a subjective scale depending on the perception of people.

Percentage of audio with the original one (%)	Evaluation	MSE	MAE
90 %	3	0.01381	0.08727
80 %	3	0.00785	0.05920
70 %	3	0.00445	0.04061
60 %	3	0.00237	0.02677
50 %	3	0.00122	0.01800
40 %	3	0.00064	0.01324
30 %	2	0.00041	0.01065
20 %	2	0.00024	0.00866
10 %	1	0.01381	0.08727

suit the two use cases, a thorough study was made on how to create damaged audio files which resemble real-world situations. In the case of noise, it was decided to work with GWN, since it is very similar to many random processes that occur in the real world. In the case of loss of information, the audio files were damaged by eliminating part of their samples as randomly as possible. Once all the audio files were damaged and the final dataset was built by normalizing, a deep learning model was developed. Its architecture consisted of an hourglass architecture using convolutional and deconvolutional 1-dimension layers along with pooling and skip connection. After training two twin models with two different datasets, one for each use case, an accuracy of 98% was obtained. Audio files with loss of information were restored in the same percentage. These results were confirmed both objectively and subjectively.

The model can be used in real-time streaming, provided that the initial delay is acceptable to the broadcaster (either radio or television). As it stands now, the clean audio exits the model 2 s after streaming starts. This time is the duration of the input data needed by the neural model. The pre, post, and processing (split, clean, and paste) times are negligible. However, the 2-second delay may not be acceptable for a live stream, and we are working on reducing this initial lag using several strategies to be published in a future paper.

Future research could continue focusing on various tracks. For example, working with higher quality audio files and/or stereo. The main limitation of the research was the size of the data. It was not possible to make use of higher quality due to computational and time limitations. If these two factors could be improved, this study could be continued exactly as conducted thus far, but with more data, higher sampling frequency, and a greater number of channels.

Working with music. For reasons like those indicated above, this project did not work with music files because of the size and quality of the data. A possible continuation or reorientation of this study could be to create a new training dataset but with music files rather than speech or voice files. Music files can be damaged, or the model trained just as in this research.

Other noise and damage. As explained, the noise used to damage the audio files was Gaussian White Noise. A very interesting way to continue

this research would be to study different scenarios or real-world cases where another type of noise is produced and learn how to reproduce it and so damage the audio files. Thus, after training the model with this new dataset, it will most likely work in new scenarios and with new use cases.

Proposal of new use cases. For example, it would be interesting to make the same study by restoring audio that comprises both cases presented in this work: background noise and loss of information.

On the other hand, what has been achieved in this research could be continued for the creation of real applications such as streaming audio reconstruction. It has been shown that the performance of the network allows real-time reconstructions, and this could be a possible application. It would therefore be possible to improve the quality of telephone calls in emergency situations. Additionally, if it is possible to recreate the same noise or damage, it may be possible to train the model with this new case. This would allow the development of various applications such as the reconstruction of old audio files, accident recordings, etc.

CRediT authorship contribution statement

Alberto Nogales: Conceptualization, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Supervision. **Santiago Donaher:** Software, Validation, Formal analysis, Data curation. **Álvaro García-Tejedor:** Validation, Formal analysis, Investigation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Ali, S., Zhou, F., Bailey, A., Braden, B., East, J. E., Lu, X., & Rittscher, J. (2021). A deep learning framework for quality assessment and restoration in video endoscopy. *Medical Image Analysis, 68*, 101900. https://doi.org/10.1016/j.media.2020.101900

Ashraf, Y. J., & Lee, C. H. (2021). Underwater ambient-noise removing GAN based on magnitude and phase spectra. *IEEE Access, 9*, 24513–24530. https://doi.org/10.1109/ACCESS.2021.3051263

Balakrishnan, A.V. & Mazumdar, R.R. On powers of Gaussian white noise. *IEEE Transactions on Information Theory, 57*, 7629–7634. DOI: 10.1109/TIT.2011.2158062.

Bengio, Y., Courville, A. C., Goodfellow, I. J., & Hinton, G. E. (2015). Deep learning. *Nature, 521*, 436–444.

Cabras, G., Canazza, S., Montessoro, P.L., & Rinaldo, R. (2010). The restoration of single-channel audio recordings based on non-negative matrix factorisation and perceptual suppression rule. In *Proceedings of the 13th International Conference on Digital Audio Effects-DAFx-10, Graz, Austria*.

Czyżewski, A., Kupryjanow, A., & Kostek, B. (2012). Online sound restoration for digital library applications. In R. Bembenik, L. Skonieczny, H. Rybiński, & M. Niezgodka (Eds.), *Intelligent Tools for Building a Scientific Information Platform. Studies in Computational Intelligence* (vol 390). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-24809-2_14.

Deng, J., Schuller, B., Eyben, F., Schuller, D., Zhang, Z., Francois, H., & Oh, E. (2020). Exploiting time-frequency patterns with LSTM-RNNs for low-bitrate audio restoration. *Neural Computing and Applications, 32*(4), 1095–1107.

De Vylder, J., Donne, S., Luong, H., & Philips, W. (2016). Image restoration using deep learning. In *25th Belgian-Dutch Conference on Machine Learning (Benelearn)*.

Dong, C., Loy, C.C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. In Fleet, D., Pajdla, T., Schiele, B., & Tuytelaars, T. (Eds.), *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, vol. 8692. Springer, Cham. DOI: 10.1007/978-3-319-10593-2_13.

Esquef, P., Välimäki, V., & Karjalainen, M. (2001). Audio restoration using sound source modeling. In *Proceedings 2001 Finnish Signal Processing Symposium, Espoo, Finland*. DOI:10.1.1.22.5874.

Franks L. & Witt, F. (1960). Solid-state sampled-data bandpass filters. In *IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. DOI: 10.1109/ISSCC.1960.1157262.

Godsill, S., Rayner, P., & Cappé, O. (2002). Digital audio restoration. In *Applications of digital signal processing to audio and acoustics* (pp. 133–194). Boston, MA: Springer. https://doi.org/10.1007/978-1-4471-1561-8.

Hsieh, T. A., Wang, H. M., Lu, X., & Tsao, Y. (2020). Wavecrn: An efficient convolutional recurrent neural network for end-to-end speech enhancement. *IEEE Signal Processing Letters, 27*, 2149–2153.

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., Estève, Y. (2018). TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation. In Karpov, A., Jokisch, O., & Potapova, R. (Eds.) *Speech and Computer. SPECOM 2018. Lecture Notes in Computer Science,* vol 11096. Springer, Cham. DOI: 10.1007/978-3-319-99579-3_21.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Kanetada, N., Yamamoto, R., & Mizumachi, M. (2013). Evaluation of sound quality of high resolution audio. *The Japanese Journal of the Institute of Industrial Applications Engineers, 1*(2), 52–57. https://doi.org/10.12792/jjiiae.001.02.003

Karimov, A., Razumov, A., Manbatchurina, R., Simonova, K., Donets, I., Vlasova, A., & Ushenin, K. (2019). Comparison of UNet, ENet, and BoxENet for Segmentation of Mast Cells in Scans of Histological Slices. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON),* pp. 0544-0547. DOI: 10.1109/SIBIRCON48586.2019.8958121.

Kawabe, H., Shimomura, Y., Nambo, H., Seto, S. (2019). Application of deep learning to classification of braille dot for restoration of old braille books. In Xu, J., Cooke, F., Gen, M., & Ahmed, S. (Eds.), *Proceedings of the Twelfth International Conference on Management Science and Engineering Management. ICMSEM 2018. Lecture Notes on Multidisciplinary Industrial Engineering.* Springer, Cham. DOI:10.1007/978-3-319-93351-1_72.

Khan, N. M., & Khan, G. M. (2021). Multi-chromosomal CGP-evolved RNN for signal reconstruction. *Neural Computer & Applications, 33*, 13265–13285. https://doi.org/10.1007/s00521-021-05953-4

Kingma, D.P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimisation.* CoRR, abs/1412.6980.

Kipnis, A., Goldsmith, A. J., & Eldar, Y. C. (2015). Optimal trade-off between sampling rate and quantization precision in Sigma-Delta A/D conversion. *International Conference on Sampling Theory and Applications (SampTA), 2015*, 627–631. https://doi.org/10.1109/SAMPTA.2015.7148967

Kiranyaz, S., Ince, T., Hamila, R., & Gabbouj, M. (2015, August). (2015). Convolutional Neural Networks for patient-specific ECG classification. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2608-2611. DOI: 10.1109/EMBC.2015.7318926.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84–90. https://doi.org/10.1145/3065386

Godsill, S. J., Wolfe, P. J., & Fong, W. N. W. (2001). Statistical model-based approaches to audio restoration and analysis. *Journal of New Music Research, 30*(4), 323–338. https://doi.org/10.1076/jnmr.30.4.323.7489

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. In *Lecture Notes in Computer Science* (vol 1681)Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-46805-6_19.

Lim, J., & Oppenheim, A. (1978). All-pole modeling of degraded speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 26*(3), 197–210. https://doi.org/10.1109/TASSP.1978.1163086

Liu, P.-Y. & Lam, E. Y. (2018). Image Reconstruction Using Deep Learning. CoRR, abs/1809.10410.

Lu, L., Mao, Y., Wenyin, L., & Zhang, H. J. (2003, July). Audio restoration by constrained audio texture synthesis. In *International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, 2003, pp. III-405. DOI: 10.1109/ICME.2003.1221334.

Lu, X., Tsao, Y., Matsuda, S., & Hori, C. (2013, August). Speech enhancement based on deep denoising autoencoder. In *Interspeech* (pp. 436-440).

Mathai, M. K., & Deepa, J. (2015). December). Design and implementation of restoration techniques for audio denoising applications. In *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)* (pp. 21–26). https://doi.org/10.1109/RAICS.2015.7488382

Menendez-Ortiz, A., Feregrino-Uribe, C., & Garcia-Hernandez, J. J. (2018). Self-recovery scheme for audio restoration using auditory masking. *PloS one, 13*(9), e0204442.

Miura, S., Nakajima, H., Miyabe, S., Makino, S., Yamada, T., & Nakadai, K. (2011, November). TENCON 2011 - 2011 IEEE Region 10 Conference, 2011, pp. 394-397, DOI: 10.1109/TENCON.2011.6129132.

Náplava, J., Straka, M., Stranák, P., & Hajic, J. (2018). Diacritics Restoration Using Neural Networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA)*.

Narayanaswamy, V. S., Thiagarajan, J. J., & Spanias, A. (2021). On the Design of Deep Priors for Unsupervised Audio Restoration. arXiv preprint arXiv:2104.07161.

O'Shea, T. J., Roy, T., & Clancy, T. C. (2018). Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing, 12*(1), 168–179. https://doi.org/10.1109/JSTSP.2018.2797022

Othman, H. A. (2016). Generalised free Gaussian white noise. *Mathematics, 8*(6), 1025. https://doi.org/10.3390/math8061025

Pandey, A., & Wang, D. (2019). A new framework for CNN-based speech enhancement in the time domain. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 27, no. 7, pp. 1179-1188, July 2019, DOI: 10.1109/TASLP.2019.2913512.

Pascual, S., Bonafonte, A., & Serra, J. (2017). SEGAN: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*.

Pras, A., & Guastavino, C. (2010). Sampling Rate Discrimination: 44.1 kHz vs. 88.2 kHz. *Audio Engineering Society Convention 128*. Audio Engineering Society.

Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C. H. (2020). On mean absolute error for deep neural network based vector-to-vector regression. In *IEEE Signal Processing Letters,* vol. 27, pp. 1485-1489, 2020, DOI: 10.1109/LSP.2020.3016837.

Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W., & Frangi, A. (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science*, vol 9351. Springer, Cham. DOI: 10.1007/978-3-319-24574-4_28.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Salloum, W., Finley, G., Edwards, E., Miller, M.A., & Suendermann-Left, D. (2017). Deep Learning for Punctuation Restoration in Medical Reports. In *BioNLP* 2017, pages 159–164, Vancouver, Canada. Association for Computational Linguistics.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualisation. *Human Brain Mapping, 38*(11), 5391–5420. https://doi.org/10.1002/hbm.23730

Siegert, I., Lotz, A. F., Duong, L. L. & Wendemuth, A. (2016). Measuring the impact of audio compression on the spectral quality of speech data. In *Elektronische Sprachsignalverarbeitung 2016. Tagungsband der 27. Konferenz* (p./pp. 229–236), Leipzig, Germany: TUDpress.

Sola, J., & Sevilla, J. (1997). Importance of input data normalisation for the application of neural networks to complex industrial problems. *In IEEE Transactions on Nuclear Science, 44*(3), 1464–1468. https://doi.org/10.1109/23.589532

Sprechmann, P., Bronstein, A., Morel, J. M., & Sapiro, G. (2013). May). Audio restoration from multiple copies. *IEEE International Conference on Acoustics, Speech and Signal Processing, 2013*, 878–882. https://doi.org/10.1109/ICASSP.2013.6637774

Stoller, D., Ewert, S., & Dixon, S. (2018). Wave-u-net: A multi-scale neural network for end-to-end audio source separation. arXiv preprint arXiv:1806.03185.

Thon, L. E. (2003). 50 years of signal processing at ISSCC. 2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. *San Francisco, CA, USA, 2003*, 27–28. https://doi.org/10.1109/ISSCC.2003.1264034

Trivedi, A., Pant, N., Shah, P., Sonik, S., & Agrawal, S. (2018). Speech to text and text to speech recognition systems-A review. *Journal of Computer Engineering, 20*(2), 36–43. https://doi.org/10.9790/0661-2002013643

Yeh, C. H., Huang, C. H., Kang, L. W., & Lin, M. H. (2018). November). Single image dehazing via deep learning-based image restoration. *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018*, 1609–1615. https://doi.org/10.23919/APSIPA.2018.8659733

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2018). Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9446–9454).

Venkataramani, S., Higa, R., & Smaragdis, P. (2018, November). Performance based cost functions for end-to-end speech separation. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 350–355). IEEE.

Zhang, K., Zuo, W., Gu, S., & Zhang, L. (2017). Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3929–3938).