DUBLIN CITY UNIVERSITY
SCHOOL OF ELECTRONIC ENGINEERING

Using Curriculum Learning to transmit images over the air

Pau Bernat i Rodríguez

May 2022

BACHELOR OF ENGINEERING

IN

DATA SCIENCE AND ENGINEERING

Supervised by Dr. Kevin McGuiness

Acknowledgements

Firstly, I would like to thank my supervisor Dr. Kevin McGuinness for the opportunity of working in a project like this, and for the guidance provided during the semester. Secondly, I would like to thank my two advisors Dr. Jordi Pons and Dr. Xavier Giró-i-Nieto, for encouraging me to do my best every week and for supporting this work. I would also like to thank Rita Geleta for starting this amazing project and also thank Tere Domenech and Cristina Puntí for the help they provided whenever it was needed. I would also like to acknowledge my colleague and friend Jaume Ros, who saved me hours of debugging with his knowledge of python and his large patience. Lastly, I would like to thank my parents and my brother, that even though they were far away, they encouraged me to do the best thesis possible.

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

I have read and understood the DCU Academic Integrity and Plagiarism at https://www.dcu.ie/system/files/2020-09/1_-_integrity_and_plagiarism_policy_ovpaa-v4.pdf and IEEE referencing guidelines found at https://loop.dcu.ie/mod/url/view.php?id=1401800 .

Name:PAU BERNAT I RODRÍGUEZ Date:13/05/2022

Abstract

Steganography is the practice of concealing a signal inside another signal. In this work, we use a modification of the original PixInWav, a deep steganography multimodal system that hides images inside of audio, to approach a more complex over-the-air transmission problem, where the audio with a hidden image concealed is reproduced through a speaker, recorded by a microphone, and sent as input to the decoder network. To tackle this problem, we use Curriculum Learning and Data Augmentation progressively adding reverberation to the training, gradually increasing the difficulty of the entries fed to the network. This new approach obtained very promising results both in terms of image and audio quality, although it remains pending to assess this performance in a real over-the-air transmission scenario.

Table of Contents

Table of Figures

Chapter 1 - Introduction

Steganography often is described as the field of concealing a signal within another signal, where the first signal, often referred to as the host signal, is usually publicly available, whereas the second signal, referred to as the hidden signal, is secretly embedded in the host signal. Throughout history, many different techniques have been used to hide signals inside of others and we can distinguish two main types of techniques: physical and digital. On the one hand, the physical steganography techniques have been used for centuries and are based on concealing the hidden signal into a physical host signal. Some examples of physical steganography are writing messages on paper using secret inks, the usage of photographically-produced microdots, or messages written in morse code in yarn and then knitted into a clothing piece. On the other hand, the digital steganography techniques relied on concealing a digital signal inside of another. Specifically, in the field of image, digital steganography often relied on the least significant bit approaches [1], where the least significant bits are used to conceal the hidden image.

The recent uptrend and advances in deep learning, have caused great improvements in the state of the art of the field of digital steganography, similar to many other fields such as image compression or audio denoising. These steganography systems are based on encoder-decoder setups, where an encoder network conceals a signal inside of another and a decoder network retrieves the original hidden from the output of the encoder network (referred to as the container signal). In the image field, deep learning systems like StegaStamp [2], which concealed hyperlinks inside of images, or HiDDen [3] - see Fig. 1, which concealed images inside of images, showed the state of the art results not only in terms of the quality of recovered hidden signals but also in the imperceptibility of these when hidden in the host signal. The same trend occurred in the field of audio, where systems like "Hide and Speak" [4], which hid speech signals within speech signals, showed again state of the art results both in recovered hidden audio quality, but also in the imperceptibility of the hidden audio when concealed inside the host audio.

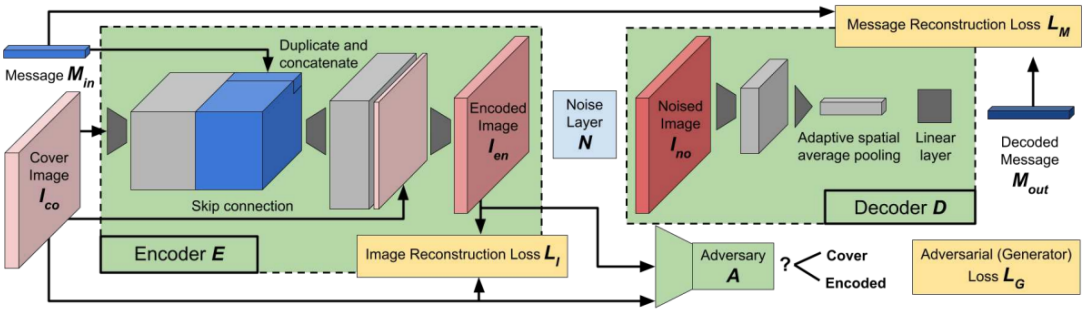

Figure 1: HiDDen steganography architecture proposed in Zhu et. al [3]

These deep-learning-based systems focused either on unimodal steganography, that is, hiding a signal within a signal of the same type, or on hiding a fairly simple signal, such as the plain text of a link, within another more complex signal, such as audio, image or video. PixInWav [5] (Fig. 2) was proposed to hide image signals inside of audio signals, focusing on the not explored field of multimodal steganography based on deep learning systems. This system, based on the usage of two UNet-like [6] networks as the encoder and decoder, relied on the transformation of the audio into the Short-Time Discrete Cosine Transform domain before concealing independently encoded images into this host audio representation.

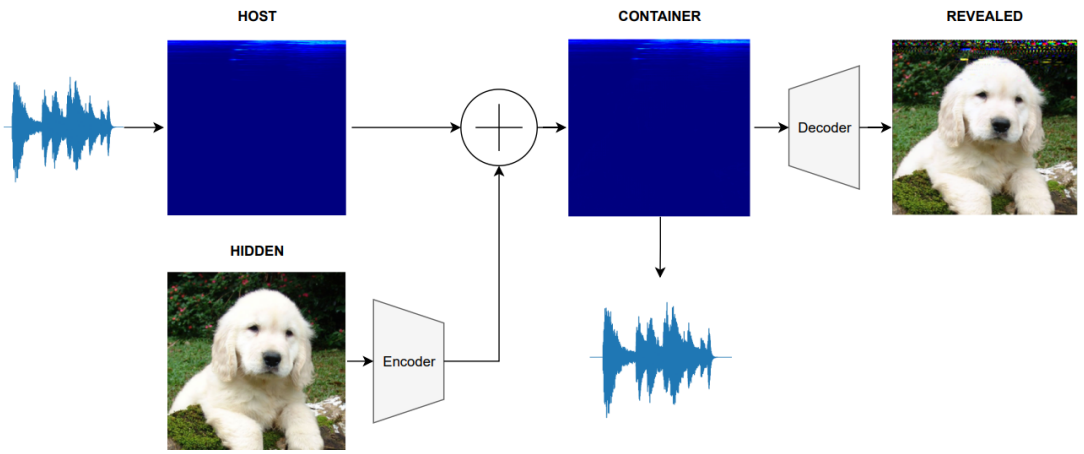

Figure 2: Original PixInWav architecture proposed in Geleta et. al [5]

One of the main problems that this system encountered was its usage of it in a real case scenario, the over-the-air transmission problem (Fig. 3), where audio that conceals a signal (previously encoded by the encoder part of the network) is reproduced on a speaker and someone wants to recover the hidden image by recording this audio and passing it through the decoder part of the network. The experiments performed showed that the network wasn't

robust enough to decode the image from this reproduced audio, returning an almost completely grey image with no correlation to the original hidden image.

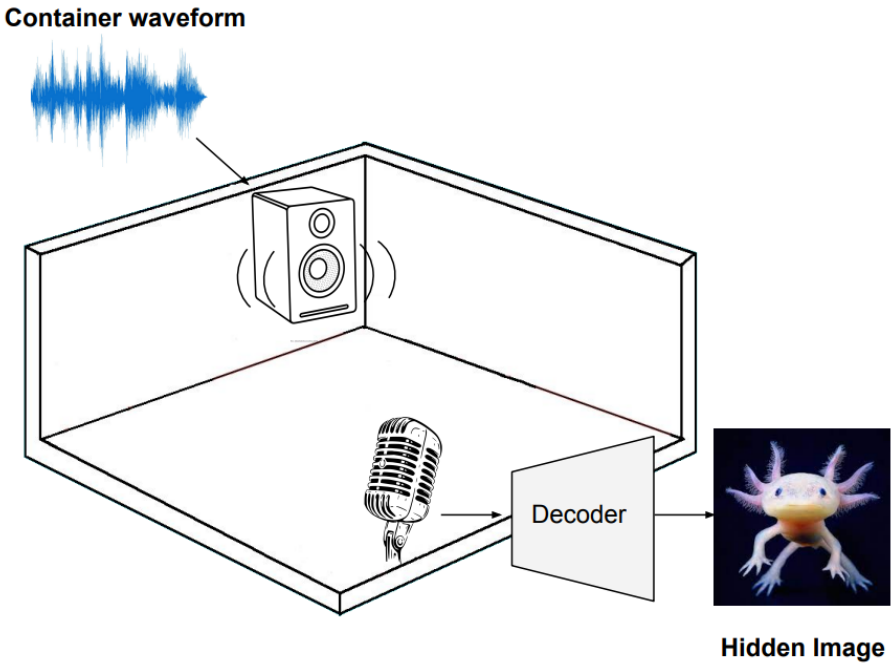

Figure 3: Over-the-air transmission. The container audio is reproduced by a speaker that a microphone records and sends to the decoder network, that reveals the hidden image

To tackle this over-the-air transmission problem, we decided to model the effect of the reproduction by a speaker in a room as the convolution of our container signal with the impulse response of the environment. It is important to mention that using this simplification of the problem, we were not modelling the effects that the speaker and recorder produce on the audio. We chose this strategy since it can be considerably complex to simulate the effect of all the possible speakers and the recorders, given that the effect of these devices on the audio varies drastically when changing them. Instead of that, we used a Data Augmentation/Curriculum Learning approach, where during the training we progressively reverberated the container signal with environmental impulse responses, increasing progressively the "difficulty" of the reverbs fed to the network

Chapter 2 - Technical Background

In this section, we will introduce in detail some of the important concepts necessary to understand the main contributions of this project, considering also related works that validate our approach to the over-the-air transmission problem. To this end, in the following subsections, we will explain audio transforms (and which ones are being used), Environmental Impulse Responses (EIRs), Data Augmentation, Curriculum Learning, and lastly, the original PixInWav [5] system.

2.1. Audio transforms:

The PixInWav original system is based on the transformation of the audio waveform into a two-dimensional signal using the Short-Time Discrete Cosine Transform (STDCT), which later was replaced by the Short-Time Fourier Transform (STFT) because of the increase in terms of quality in the container audio. In this section, we will briefly introduce both transforms.

2.1.1 Short-Time Discrete Cosine Transform (STDCT)

The STDCT of a signal is a transform that computes the DCT transform (specifically the DCT-II, commonly referred as the DCT) for L-sized windows of the original signal. The mentioned DCT expresses a finite signal using a sum of cosine signals of different frequencies and it is widely used in the field of audio processing and compression. The formula to compute the STDCT is:

$$X_{STDCT}[m,n] = \sum_{k=0}^{L-1} x[k]g[k-m]cos(\frac{\pi}{L}(n+\frac{1}{2})k)$$

Where $x[k]$ is our original audio signal and $g[k]$ is an L-point window function. The resulting signal after the transformation is a 2-D signal that belongs to the real domain. Lastly, there is an inverse transformation that allows the recovery of the original signal from its STDCT representation.

2.1.2 Short-Time Fourier Transform (STFT)

The STFT transformation is a natural extension of the Fourier Transform, providing Fourier Transforms of a windowed signal. It is widely used in the audio field to extract the time-localised frequency information of the audio and it is computed using the following formula:

$$X_{STFT}[m,n] = \sum_{k=0}^{L-1} x[k]g[k-m]e^{-j2\pi nk/L}$$

Where again, $x[k]$ is our original audio signal and $g[k]$ is an L-point window function. The resulting signal after the transformation is a 2-D signal that belongs to the complex domain, which means that we can split this signal in two parts: its module and its phase. Similar to the STDCT, there is also an inverse transformation that allows the recovery of the original signal from its STFT representation.

2.2. Environmental impulse responses:

In order to simulate the effect of a speaker reproducing the container audio, we decided to use the MIT Acoustical Reverberation Scene Statistics Survey [6] dataset of environmental impulse responses. An impulse response is the output of a system when presented with an impulse signal. In the acoustics field, room impulse responses are widely used as they describe the characteristics of a specific enclosed location. Environmental impulse responses (EIRs) are a generalisation of the room impulse response concept since it also considers other spaces apart from rooms (like open spaces). In our practical case, we used these impulsional responses to simulate the effect of the environment on our container signal by simply performing a discrete convolution of this signal and with environmental impulse responses (or reverbs).

The usage of reverberations (or impulse responses) in the speech enhancement field is a usual practice since it serves a way to approach the very complex problem of removing the effects of the room on speech. In studies like [7] where they use perceptual metrics to perform speech recognition or in [8] where they perform speech recognition from whispering speech, they approach these problems by adding reverberation to the training data (simultaneously with other kinds of noises and artefacts). In the state-of-the-art

approach to the speech enhancement task [9] (fig.4), where they estimate the lip movement (without video clues) from an audio signal, the usage of reverberations is still used.

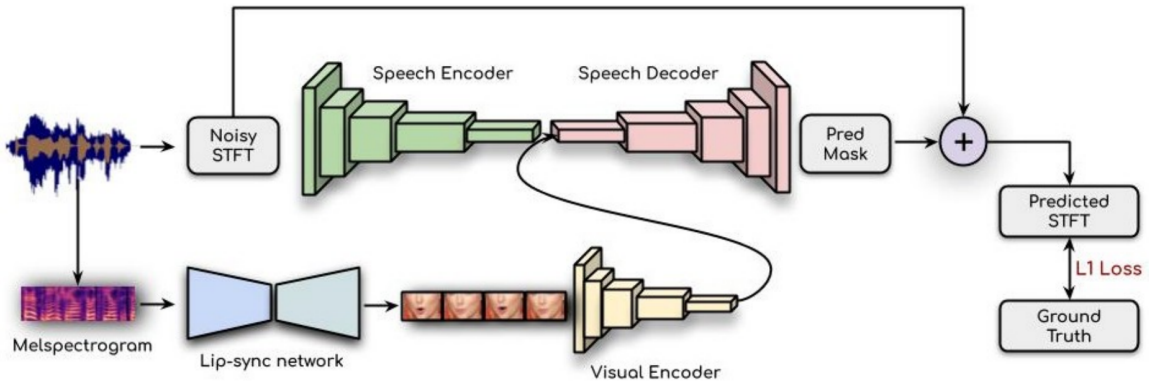

Figure 4: State-of-the-art architecture in the speech enhancement task proposed in Hedge et. al [10]

We also considered some interesting literature from the impulsional response estimation field like FiNS [10] (Fig. 5), where a more domain-inspired architecture uses noise filtering to estimate impulse responses, or systems like the ones presented in [11] and [12],which estimate the room impulse responses by using images of the places where the audio was recorded. We left as possible future work the option of estimating the impulse response of the room to eliminate the effects of it in the transmitted container audio.

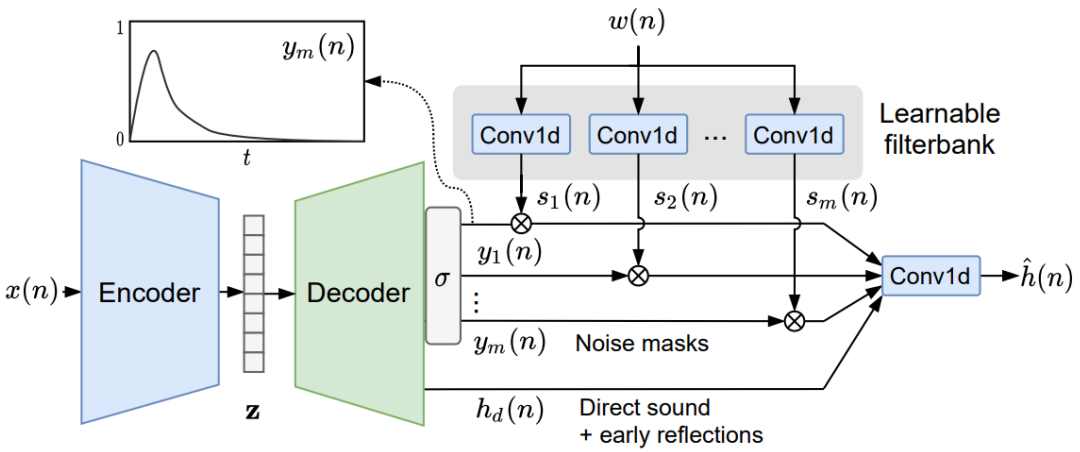

Figure 5: FiNS, a domain-inspired architecture for room impulse response estimation proposed in Steinmetz et. al [10]

2.3. Data augmentation:

Data augmentation is a widely used technique in the machine learning field that is based on increasing the size of your dataset by artificially generating extra data, often using transformations on the original data. However, transformations are not the only option to generate artificial data since other approaches like using GANs to generate these images [13] have been also used in the recent literature.

The aim of data augmentation in our specific practical case is not to obtain more data of the same original distribution, but to modify the distribution of our data in a way that each of the training entries is affected to some degree by the reverberation effect caused by the acoustical environment. As mentioned before, this approach has been used previously in the literature, specifically in the speech enhancement field [7,8,9] where models learn how to clean the audio signals from all kinds of undesired effects like noise, reverberation or quantization effects by training with these applied to the data. Therefore, we will use data augmentation to make our network robust to the effect of reverberation caused by the environment.

2.4. Curriculum learning:

Curriculum learning is a concept firstly introduced by Bengio et al. (2009) [14] which suggests that machine learning models, similar to humans or animals, learn better when examples are presented in a meaningful order, that is, increasing progressively the difficulty of the entries. Specifically, the curricular learning method introduced in this paper was the following: First, they trained the model by giving a set of weights that favoured sampling the "easier" cases of a dataset. Next, they progressively modified these weights so more "hard" cases were sampled. At the end of this process, the weights didn't favour any kind of entry, and sampling any case was equally probable despite its difficulty.

This far from new concept has been used and adapted in a wide variety of cases. The paper "Curriculum Learning: A Survey" [15], summarises the extensive list of cases and variations of Curriculum Learning in literature, where the progressive increase of difficulty of the training is generalised into a wider frame, where the difficulty of the samples can be self-learned or where other factors like diversity can also influence the sampling probability of an entry.

There is some work in the audio field where the use of Curriculum Learning improves the performance of the models. Specifically, in the paper [16] from the speech denoising field, they use a Curriculum Learning method called ACAM where, during the training, the entries sampled have progressively a higher signal to noise ratio (SNR). In [17], from the speech recognition framework, they fed progressively harder sentences to the network.

In our specific case, we want to train progressively increasing the difficulty of the reverbs fed to the network. To this end, in the following subsection, we will introduce the dry-wet parameter concept, which will be used as the main source to define how "difficult" is an entry.

2.5. Dry-Wet ratio:

The dry-wet ratio is a parameter used by most of the reverberation software available in the market that determines how much of the original audio is heard after a reverberation. Thus, the audio after the μ dry-wet parameter reverberation is computed as:

$$x_{\mu-reverbered}[n] = \mu \cdot x[n] * r[n] + (1-\mu) \cdot x[n]$$

Where x[n] is our original signal, r[n] is the reverb signal and x[n] * r[n] is the fully reverberated signal. Thus, when the μ dry-wet parameter is close to 1 the returned signal is more similar to the fully reverberated signal and when the μ dry-wet parameter is close to 0 the returned audio is more similar to the original audio.

As we observed in the previous PixInWav experiments, the system was incapable of decoding the hidden image when the container was reverberated (or reproduced-recorded). Then, it is not bold to assume that, if we use a dry-wet parameter reverberation approach to train our network, the entries of the database with a lower dry-wet parameter on their reverberated container are going to be easier to decode for the network. Combining this knowledge with the Curriculum Learning framework mentioned above, we can schedule a training method where first, we sample examples with lower dry-wet parameters (easier examples) and we increasingly add samples to the pool with higher dry-wet parameters (harder examples) until all samples are chosen with the same probability.

2.6. PixInWav:

The PixInWav [5] system is a multimodal steganography system that encodes images inside of audio. It is based on an encoder-decoder setup, where a UNet-like [18] residual architecture encodes the image (hidden image) independently from the audio and this encoded signal is added to the Short-Time Fourier Transform (STFT) module (of the host audio. Then, the resulting audio, also called container audio, is decoded by another UNet-like (Fig. 6) network that retrieves the hidden image from this container audio. In this section, we will introduce in more detail how this deep neural network system works, the previous results of this system and the improvements added since the original PixInWav paper that are not related to the reproduction-recording problem.

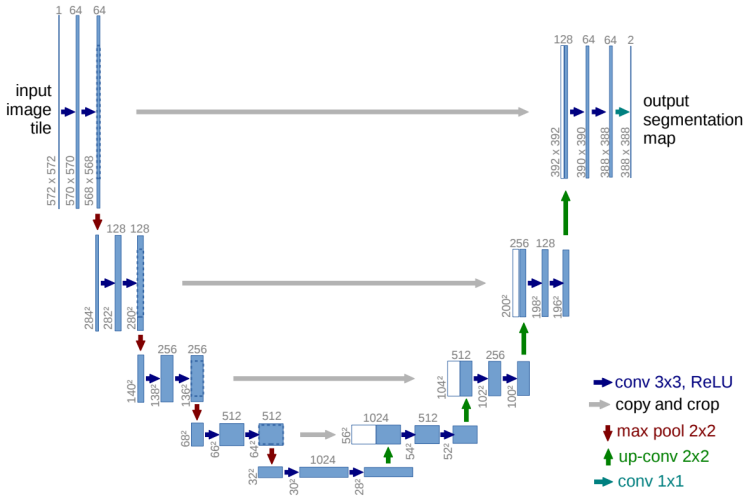

Figure 6: U-Net architecture, proposed in Ronneberger et. al [14]

2.6.1. Architecture

The PixInWav system follows an end-to-end encoder-decoder residual architecture based on two UNet-like [18] networks. Differently from other mentioned deep stenography systems, in PixInWav the hidden signal (image) is encoded independently of the host signal (audio). This is why the encoder uses as input only a 3-channel RGB 256x256 image, that after a series of reversible transformations that flatten and stretch the image, is used as the only input of the encoder network. This architecture contains both a contracting and expansive part with residual connections.

The contracting part of the encoder network is composed of two downsampling sections, with two 3×3 convolutions with stride 2 on the first module, and two 3×3 convolutions with stride 4 on the second one. Each of these layers was followed by a batch normalisation layer and a Leaky ReLU activation function. The expansive part of the encoder is composed of two upsampling modules, with two 3x3 transposed convolutional layers (upsampling) and two 3x3 convolutional layers each one followed by a batch normalisation layer. As in the contracting part, both of these layers are followed by a Leaky ReLU activation function in between. The output of this network returns the encoded hidden image, computed independently from the host audio. The decoder part of the network uses the same architecture as the encoder, only that this network receives as input the container host signal and retrieves the original hidden image

Lastly, it is important to mention the connection between both networks, that is, how the encoded hidden image is hidden inside the host audio. To do this, first, the Short-Time Discrete Cosine transform of the audio is computed to transform the 1-dimensional signal into a 2-dimensional representation of the audio, where the stride and dimensions of the window are adjusted so that this 2x2 dimensional representation matches the dimensions of the encoded hidden image. After this, we perform a simple addition of both signals obtaining the container audio representation, which then is passed through the decoder to retrieve the original hidden image.

2.6.2. Loss function

The loss function of the PixInWav system was:

$$\mathcal{L}(s, s', C, C') = \beta\|s-s'\|_1+(1-\beta)\|C-C'\|_2.+\lambda\ L1(c, c')$$

Where s is the hidden image, s' is the revealed image, C is the host audio representation, C' is the container audio representation, c is the host waveform and c' is the container waveform.

From this loss function, it is important to mention the trade-off between (i) low distortion of container audio and (ii) low distortion of the revealed image, which is controlled by the β hyperparameter. When this β value is close to 1, we prioritise the low distortion of the

recovered image whereas when the β value is close to 0, we prioritise the low distortion of the container audio.

2.6.3. Audio transforms

The Short-Time Discrete Cosine Transform (STDCT) (Fig. 7) was replaced by the Short-Time Discrete Fourier Transform (STFT) (Fig. 8) in the PixInWav system. The main reason for this change was that it produced a large increase in terms of container audio quality, at the expense of a slight loss in terms of revealed image quality. It is worth mentioning that, since the STFT transformation of an audio signal is a 2-dimensional complex signal, in this new system we added the encoded image into the module of the signal (leaving the phase untouched).

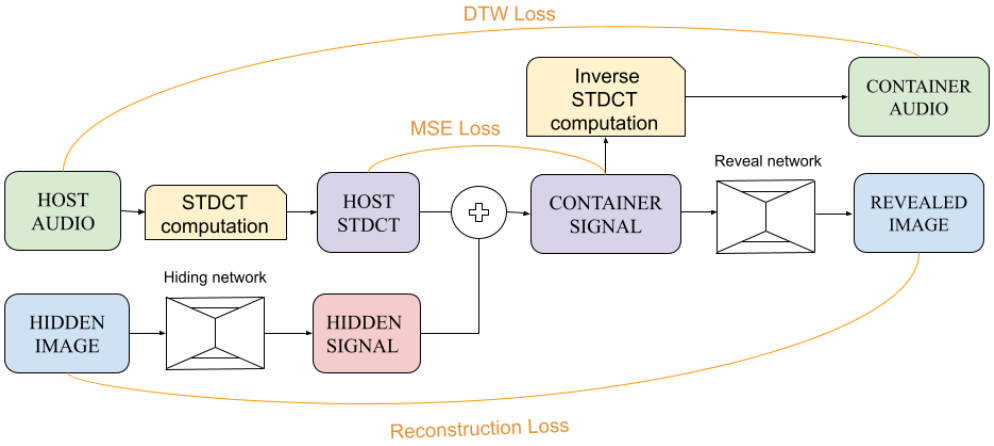

Fig 7: PixInWav original architecture using the STDCT, proposed in Geleta et. al

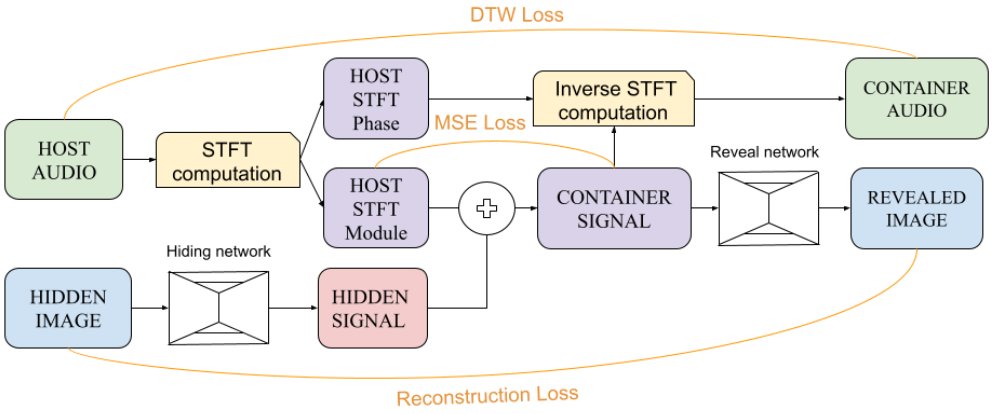

Fig 8: PixInWav modification using the STFT module as the host signal

2.6.4. Results of the original system:

In this last subsection, we will introduce some of the previous results obtained by the original PixInWav architecture, as well as the results in the STDCT vs STFT study, and lastly the baseline results in the Over-the-air transmission problem.

Main results:

In the original PixInWav paper where they used the STDCT they obtained considerably good results, obtaining structural similarity index measure (SSIM) values around 0.94 for the revealed images and Signal-to-Noise Ratio (SNR) values around 19dB for the audios, with 8 epochs of training. As seen in Fig. 9, the network was capable of recovering the hidden image from the container audios, with a quality that would be almost perfect if it were not for the marks that the spectrogram leaves in the upper part of the image.

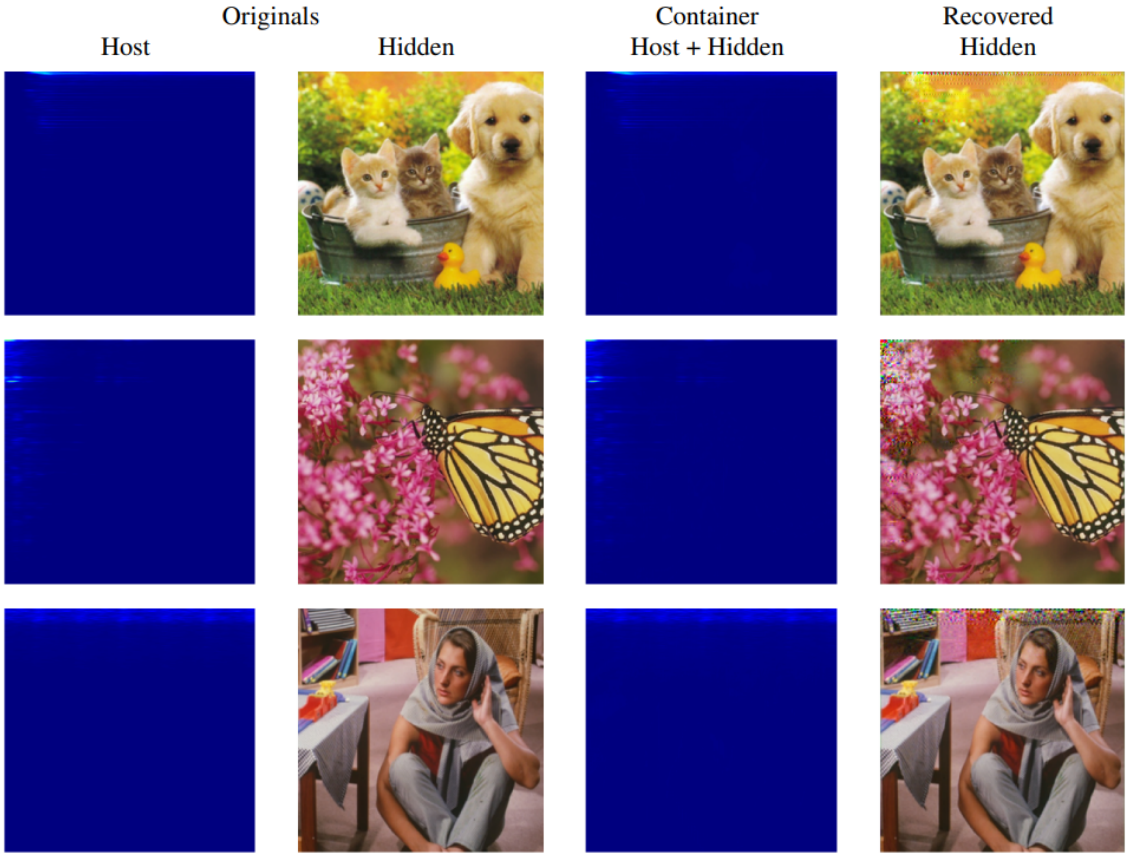

Figure 9: PixInWav original results using the STDCT, presented Geleta et. al [2]

In terms of perceptual audio quality, this method using STDCT introduced a noise frequency stationary noise to the container audio signals, which greatly deteriorated the quality of the container signal.

Short-Time Fourier Transform vs Short-Time Discrete Cosine Transform :

To obtain a better performance in terms of audio quality, the usage of the Short-Time Discrete Fourier Transform was proposed. This transform was chosen because of two main reasons: (i) the fact that we could hide on one of the two parts of the complex signal (module and phase) and use the less sensible part to noise to add the encoded image, and (ii) the fact that the usage of the STFT was the most common transform used in the audio field.

Using the module of the STFT as the container signal, we obtained an increase in terms of audio quality of over 20dBs on the SNR audio metric, obtaining a cleaner container audio without the previously mentioned artefacts at the expense of a slight degradation of the quality of the revealed image, dropping .05 in terms of the SSIM image metric for most β values (Fig. 10). In perceptual terms (Fig. 11), the degradation of the revealed image quality was specially remarkable in the experiments using the STFT with the lower beta values, but as the beta value increased this method obtained a similar revealed image perceptual quality to the STDCT method. Due to this increase in terms of performance, further PixInWav experiments were performed using the STFT transform with β values higher than 0.25.

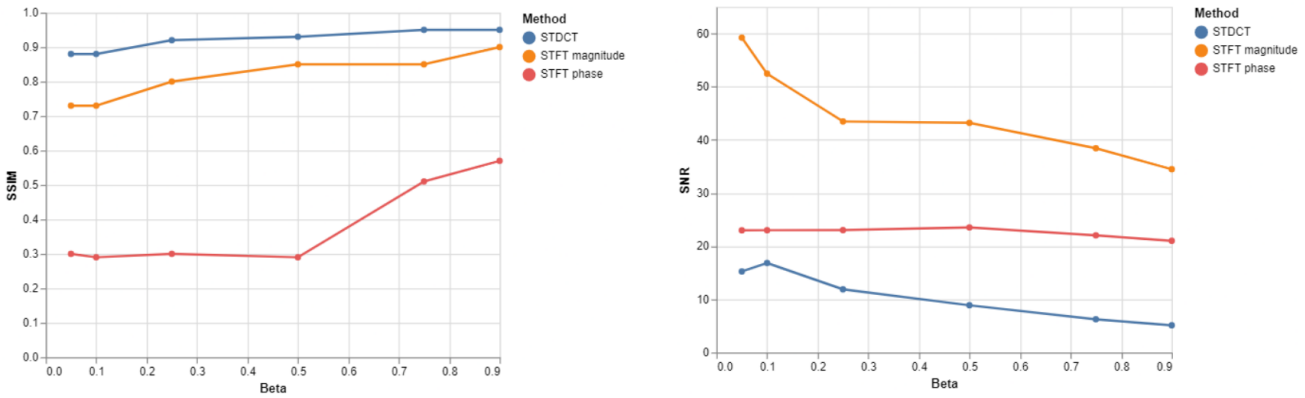

Figure 10: Visualisation of the quantitative analysis of the STFT vs STDCT

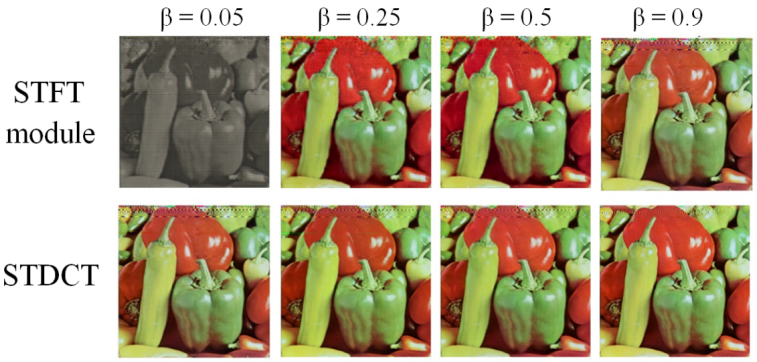

Figure 11: Perceptual results comparison of the PixInWav systems using STFT or STDCT

Over-the-air problem:

The original over-the-air experiments were conducted in the following way: a previously trained PixInWav network encoded an image inside of an audio, which was reproduced through a speaker and recorded with a microphone. The STFT module of this recording was fed to the decoder network where the hidden image was meant to be retrieved.

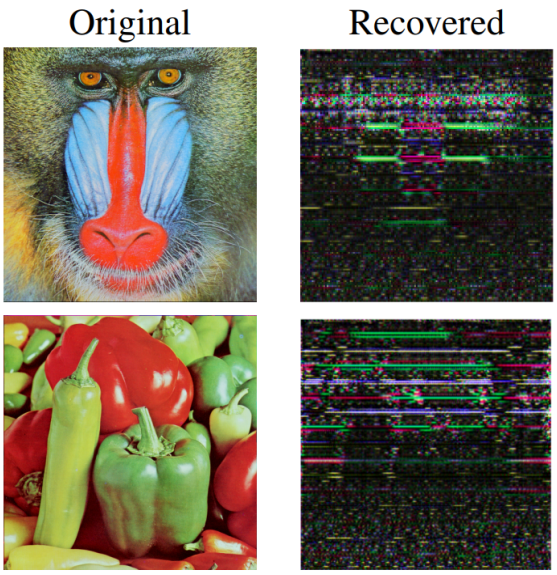

Figure 12: Results obtained in the over-the-air transmission case
using the original PixInWav system presented in et. al Geleta

As shown in Fig. 12, the results obtained were considerably poor, but some contours of the original images were preserved (like some of the peppers contours or the parts of the nose of the mandrill). In this thesis, we will focus on improving these results.

Chapter 3 - Design of the System

In this section, we will mainly focus on the design decisions used to approach the Over-the-air problem, as well as some other modifications (previously mentioned) that improve the performance of the system independently of the over-the-air case.

3.1 Over-the-air modifications:

In this section, we will introduce the modifications made to the PixInWav [5] original architecture (Fig. 10) to address the over-the-air problem. As mentioned above, we used a Data Augmentation approach to add reverberation progressively to the training dataset, using a Curriculum Learning schedule. In this new system, we added reverberations (or environmental impulse responses) to the container signal (computed as the addition of the encoded hidden image and the host audio) using the architecture shown in Fig. 13.

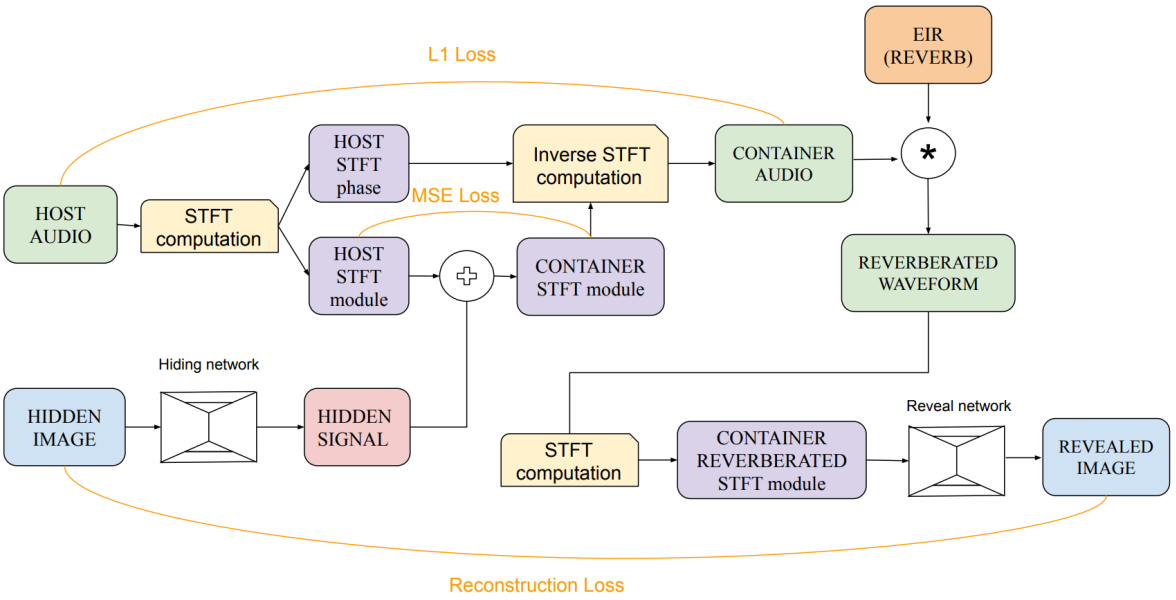

Figure 13: PixInWav architecture with reverberation in the container waveform

In this architecture, after the hidden image is encoded by the encoder network and added to the audio representation (the module of the STFT), we transform this container signal into the audio domain and compute the μ dry-wet parameter reverberation of this container. It is important to mention that we crop and standardise the fully reverberated audio so it matches the dimensions and the intensity levels of the original audio. After that, this μ-reverberated container audio is passed through the decoder network to retrieve the original hidden image.

Lastly, during training, this μ parameter is determined by the Curriculum Learning schedule, which will be explained in detail in 4.1.4, and in testing is 0 for the non-reverberated validation dataset and a random number between 0 and 1 in the reverberated validation dataset.

3.2 Audio representation:

As mentioned in the previous sections, the audio representation used in the network is the Short-Time Fourier Transform. This means that the host audio is transformed into the STFT domain before concealing the hidden image inside of it. Thus, we used this transform based on results obtained that showed a considerable improvement in the container audio quality when using the STFT transform as the transformation.

It is important to mention that since the STFT transform returns a 2-dimensional complex signal, we used the module of the signal only to hide the image. This hiding was performed by a simple linear addition operation (as mentioned before, parameters like the stride were adjusted to have matching dimensions) where the hidden image was added to the module of the audio transformation.

3.3 Increasing the network size:

The original PixInWav network had roughly 960.000 parameters. Despite being a considerably small network, the usage of a larger STDCT transform was the reason why they couldn't increase the size of this network, since it would surpass our computational limits.

However, for the over-the-air transmission task, we considered a viable option to increase the size of this network, given that the STFT transform used a smaller window. Thus, for an experiment we increased the number of the encoder and the decoder, increasing the size of the network to roughly 3.6 million parameters.

Chapter 4 - Implementing and testing

In this section, we will discuss the implementation of the design architecture, explaining also the new validation dataset, how we applied the curricular learning in the code and other modifications added to the source code. Also, we will explain the setup and design of the experiments performed and how we scheduled the Curriculum learning in those. Lastly, we will briefly discuss some other modifications performed to the original code that improved the performance, independently of the over-the-air transmission problem. The source code and our modifications were implemented in python 3.6, mainly based on the PyTorch library, but using other external libraries like NumPy, Wandb, and SciPy.

4.1. Implementation:

In the following section, we will introduce the main additions and modifications that were made to the PixInWav baseline code in order to approach the over-the-air problem. To do this, we added a reverb dataset to the code, created a second validation dataset, and added the reverberation to the architecture. We also performed some modifications to the code like changing the ratio between validation and training steps, modifying the way the training and testing databases are computed, or correcting bugs from the original code.

4.1.1. Addition of the environmental impulse responses dataset:

Since the main approach to the over-the-air problem was to add reverberation during training and validation, we added to the data loader the MIT environmental impulse responses database [6] that contains 271 reverbs sampled at 33000 Hz, where we used 230 for training and 40 for testing. We applied resampling to these reverbs to match the sampling frequency of the host audios (44100) and we extended the length of all the reverbs (with 0's) to be able to execute batch operations (which we did not use since they did not provide improvements in terms of performance of the system). We also combined these reverbs with the previous audio and image databases, where for every audio-image combination in the training or testing databases, a random reverb was matched with those (selected from its corresponding training or testing pool).

4.1.2. Creation of the reverbered validation dataset

In this subsection, I will introduce the creation of a second validation dataset where the container audios were μ-reverberated. This dataset was created to jointly validate the model with the original non-reverberated dataset since we wanted to analyse the performance of the model in the original PixInWav task and the Over-The-Air transmission task simultaneously. The reason for this addition was that we were interested in not giving up the performance in the original task to achieve better results on the Over-The-Air one.

The implementation of this dataset was fairly simple since it was based on the implementation of the original validation dataset, where the main modification was the addition of μ-reverberation (where the μ was randomly sampled from a uniform [0,1] distribution) on the container audio of each entry of the dataset. The reasoning behind this was to have all kinds of reverbs without any kind of bias based on difficulty during the validation steps.

Both validation datasets had 500 entries, and we performed a double validation step (one for each dataset) every 1000 training iterations (Fig. 10). This was a modification from the source code, where validation steps were performed every 50 training iterations. This change increased the speed of training over 10 times, which was especially useful to launch experiments with a larger number of epochs.

4.1.3. Addition of the reverberation to the model:

The addition of the reverberation in model architecture was the main change from the baseline implementation, which was added to the model script. The implementation of this addition was the following:

First, we transformed the container audio representation into the waveform domain. Then, we computed a fully reverberated signal convolving the container waveform with a random reverb. Then we cropped and standardised this fully convolved signal so it had the same dimensions and intensity as the original container audio. Then, we performed a linear combination of both signals given a randomly sampled dry-wet parameter. Regarding this

sampling, we sampled from the mu a uniform distribution (0,dry-wet limit) where this dry-wet limit was increased progressively during the training epochs until it reached its maximum value which is 1.

4.1.4. Curriculum Learning:

In this section, we will discuss how we implemented the Curriculum Learning schedule in the code, to increase the difficulty of the training progressively. As mentioned before, our mechanism to implement the progressive increase in difficulty was using the dry-wet parameter sampling range. Since entries with a lower dry-wet parameter were "easier" cases as more of the original audio was present in the reverberated signal, we implemented this Curriculum Learning by progressive increasing the sampling range values that this dry-wet parameter could take. In other words, we sampled the dry-wet parameter from a uniform distribution [0,dry-wet limit], where this limit progressively increased during training until it reached 1, which would that all possible μ dry-wet parameters are sampled with the same probability.

To implement the curriculum learning, we started with a dry-wet limit equal to 0 and from epoch 4 of training, we increased this parameter every half epoch until it reached 1. Then, we trained for a few more epochs with this maximum dry-wet limit, that is, equiprobable random sampling of dry-wet parameters, or in other words, all kinds of reverbs were sampled with the same probability.

4.1.5. Other modifications:

In this subsection, we will explain some extra modifications added to the code that improve the performance of the system, independently of the Over-The-Air transmission problem.

The first one was the reimplementation of the checkpoint system, which was supposed to save the models after we reached the limit of 24 hours in a single run established in our computation servers. This system was especially useful before changing the training-validation iterations ratio, where the experiments would often take more than 24 hours to run. The main differences from the original implementation were that in the new

implementation we also saved the optimizer parameters (apart from the model parameters) and we deleted the batch norm layers that caused problems during the saving stage.

The second one, was the modification of the DataLoader script, to avoid an overlap between the train and test audio datasets. This was a major contribution to the project, since having this overlap between both datasets meant that the validation was not independent of the training, which was an important methodology error. We also fixed another methodology error in the code, as the mode of the model was not properly changed after the first validation step. Note that after these changes, the system behaviour worsened in some aspects from the first PixInWav paper, since we are no longer training with validation entries.

The last implementation change proposed, only in some experiments, was to increase the size of the network. To do this, we simply increased the number of channels in some of the encoder and decoder layers for one specific experiment.

4.2. Setup:

In this section, we will explain with detail the metrics, datasets and hyperparameters used on the experiments performed:

4.2.1. Data sets:

We used the following datasets to conduct our experiments:

- The audio signals are the same as the ones used in the original PixInWav paper, retrieved from the FSDnoisy18K [19] dataset, an open dataset containing 42.5 hours of audio in 18,532 audio clips across 20 sound event classes, depicting a large variety of sounds, such as voice, music, or noise.

- For the images we also used the same dataset as in the original paper: we used the same sample of 10,000 256×256×3 cropped images (in the RGB space) from the ImageNet (ILSVRC2012 [20]) dataset (Fig. 10).

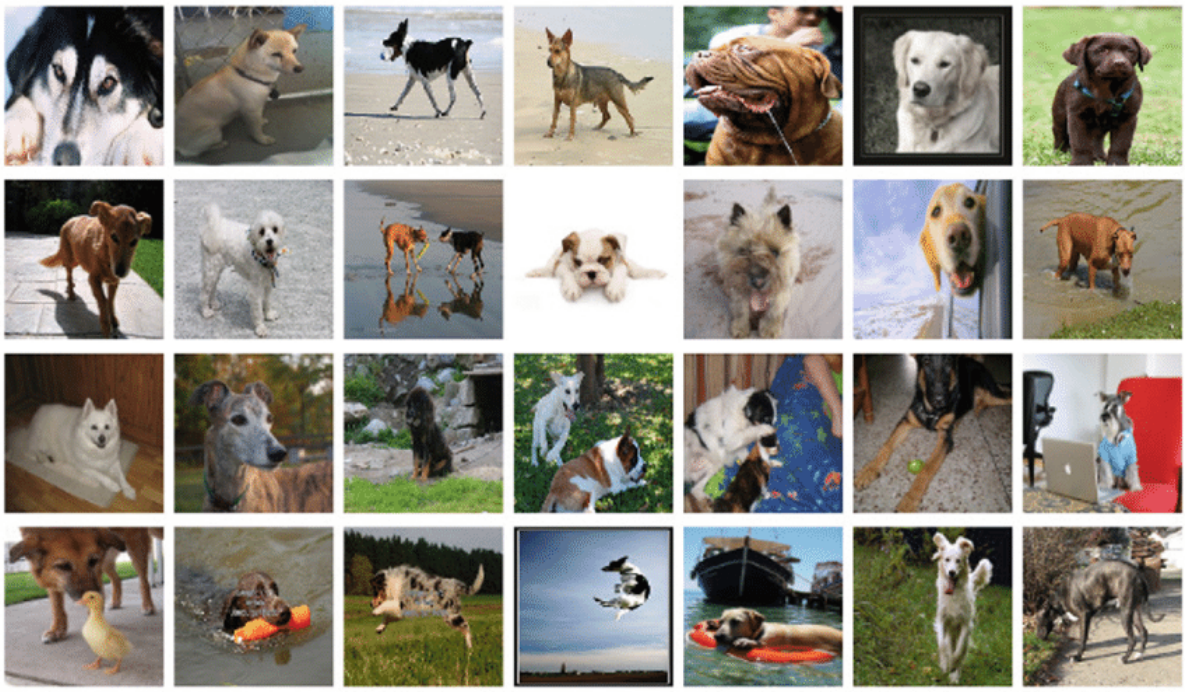

Fig 14: Dog Images from the ImageNet Database [20]

- For the environmental impulse responses (reverbs) we used the open database used in the MIT Acoustical Reverberation Scene Statistics Survey [6], containing 271 Impulse Responses measured in real-world scenarios over 301 different locations.

4.2.2. Metrics:

To measure the distortion of the container audio in relation to the host audio, we used the signal-to-noise-ratio (SNR) metric over the waveforms — where we measure the difference between the host and container audios. For the image distortion, we used the Structural Similarity Index [21](SSIM), a similarity metric for images that takes into account the human visual system, and the Peak Signal-to-Noise Ratio (PSNR) which is the ratio between the maximum power of a signal and the power of corrupting noise.

4.2.3. Training details:

For each experiment, the model was trained with the Adam optimizer at a learning rate (lr) of 0.01 and a batch size of 1 during 14 epochs. The beta parameter for the experiments was fixed at 0.9. Also, the ratio between training iterations and validation steps was changed so that for every 1000 iterations we perform a validation step of 500+500 iterations (500 per each of the two validation datasets).

4.2.4. Experiments design:

We conducted four main experiments of 14 epochs.

- An experiment where we train without any kind of reverberation in the container audio, trained to have a baseline model to perform an ablation study.

- A model where we train with reverberation but the dry-wet parameter sampled range uniformly from 0 to 1 from the beginning, that is, with no Curriculum Learning.

- A model where we train with reverberation and Curriculum Learning. In this experiment, we start training with the dry-wet parameter limit equal to 0, that is, no reverberation for the first three epochs. Then, from epoch 4 to 8 we progressively increase the dry-wet parameter limit every half epoch by 0.1, and in the last 4 epochs we train with the maximum dry-wet parameter limit, meaning that we sample all reverbs with the same probability.

- An experiment using the same setup as the previous one, but increasing the number of channels of the network.

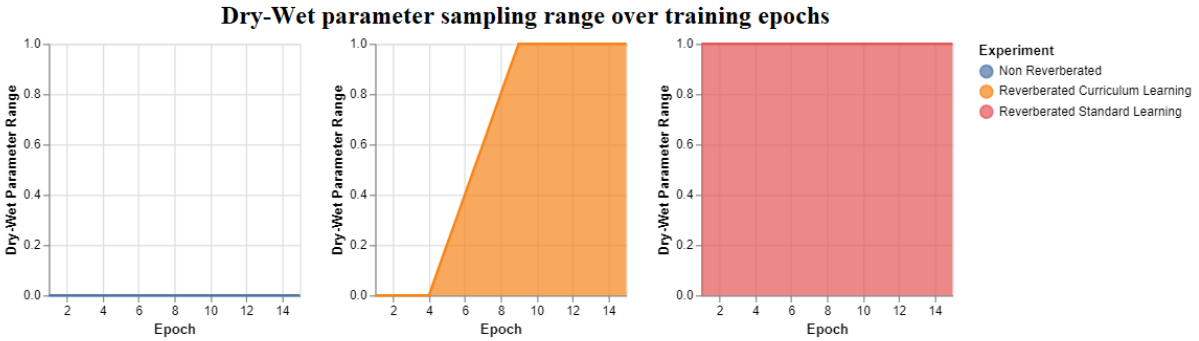

Figure 15: In this visualisation shows the evolution of the dry-wet parameter sampling range in the first three experiments over the epochs

Chapter 5 - Results and Discussion:

The results obtained in the four previous experiments are shown in Table 1 and in Fig. 16.

	Reverberated Validation Dataset			Non-Reverberated Validation Dataset		
	Audio	Image		Audio	Image	
	SNR ↑	SSIM ↑	PSNR ↑	SNR ↑	SSIM ↑	PSNR ↑
Standard PixInWav	**34.2**	0.58	14.1	**34.7**	**0.86**	**25.3**
PixInWav + Reverb	33.7	0.67	22.0	33.9	0.69	22.7
PixInWav + Reverb + Curriculum Learning	32.1	**0.75**	**22.8**	32.5	0.77	23.9
Larger PixInWav + Reverb + Curriculum Learning	34.1	0.64	20.4	33.5	0.69	21.2

Table 1: Quantitative results of the Curriculum Learning approach to the over-the-air transmission problem.

We structured the results in 4 different sections, where we analyse the performance of the Standard PixInWav model, the effectiveness of Curriculum Learning, the effects of adding reverberation to the network and the effect of increasing the size of the network. This analysis will be performed based on the previous metrics, as well as in the perceptual results in Fig. 16 and in quality of the container audios, located in the directory 'audios'.

5.1. Standard PixInWav results:

The standard PixInWav model obtains the best results with the Non-reverberated Validation Dataset, both in container audio and revealed image quality (Table 1). Despite that, we observe in Fig. 16 that we obtain worse results than in the original paper (because of the previously mentioned methodological errors). With dry-wet parameter = 0, the system only recovers the grayscale images that also have the red colour. For the higher Dry-Wet parameters (more reverberation), the returned image loses luminancy being almost black for the dry-wet parameter = 0.9.

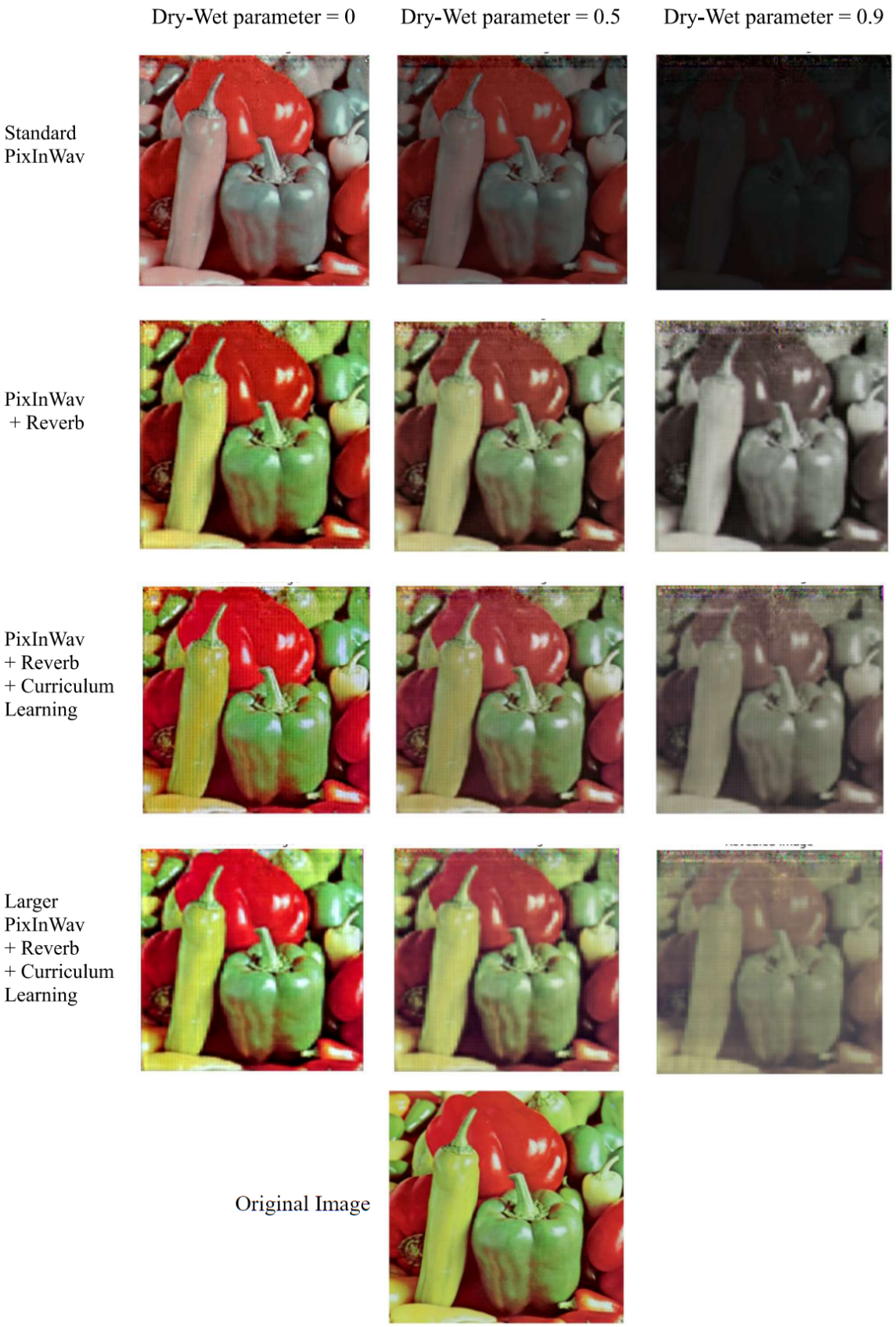

Fig. 16: Qualitative Results of the Curriculum Learning approach to the over-the-air transmission problem.

5.2. Curriculum Learning:

To assess the performance of the Curriculum Learning schedule, we are going perform an ablation study comparing the results the experiment "PixInWav + Reverberation", which uses reverberation in the training without Curriculum Learning and "PixInWav + Reverberation + Curriculum Learning", which uses reverberation and a Curriculum Learning Schedule. We obtained the following results:

- The system trained using Curriculum Learning obtains better results in terms of revealed image quality. This system obtains an increase of 0.09 in the SSIM metric and an increase of 0.8 in the PSNR in the reverberated validation data set and an increase of 0.08 in the SSIM metric and an increase 1.2 in the PSNR in the non-reverberated validation database. Perceptually, when there is no reverberation (dry-wet parameter = 0), we observe that the revealed images from the system using Curriculum Learning tend to be more similar to the original signal and when the dry-wet parameter increases the colours are betterly preserved than in the standard learning approach.

- The non-Curriculum Learning system obtained better qualitative results in terms of container audio quality for both datasets in the SNR metric. However, this difference was not perceptually noticeable as the quality of both audios was almost perfect.

Thus, using a Curriculum Learning schedule improved the performance of the model, given that we improved the image quality at the cost of a non perceptually noticeable loss in image quality.

5.3. Training with reverberation:

In this section, we will compare the results obtained by the system trained with reverberation without Curriculum Learning and the results of the original system, in both the reverebered and the non-reverberated validation datasets. We want to analyse if training with a set of reverbs convolved with the container audio improves the performance when testing with a different non-overlapping set of reverbs, as well as analysing the performance of this system in the original task, comparing with a non-reverberated training experiment of PixInWav. We obtained the following results:

- As mentioned before, the standard PixInWav model trained obtains the best results with the original validation Dataset. However, the usage Curriculum Learning is considerably useful to maintain acceptable levels of revealed image quality in the original PixInWav task, while obtaining the best results in the reverberated container task.

- The method that uses reverberation during training without a Curriculum Learning schedule, obtains better results than the original model in the reverberated validation dataset. These results are especially promising, since it means that the network has learnt how to decode container audios with reverbs that it has never seen before, which is the kind of generalisation that we needed to use these models in a real case scenario.

5.4. Increasing the size of the network:

In the last experiment that we performed we increased the size of the network by increasing the number of channels of the encoder and decoder. To assess the performance of this change, we ran an experiment with this new architecture using reverbs and a Curriculum Learning schedule, and we compared these results with the ones obtained with the smaller architecture. This larger network obtained:

- Worse quantitative and qualitative results in terms of image quality than the same experiment with the smaller network.

- An slight increase in terms of audio quality that was not perceptually noticeable.

Thus, increasing the size of the network by adding more channels didn't improve the performance of the smaller model.

5.5. Maximum reverberation case:

For the maximum reverberation case (with dry-wet parameter = 1), neither of the networks was capable of decoding the original image. It seems to be the consequence of almost never having trained with the specific case of dry-wet = 1, as we trained using continuous uniform distribution of dry-wet parameters between [0,1]. All the tested models retrieve black/grey images in this case without any trace of the original image (Fig. 17).

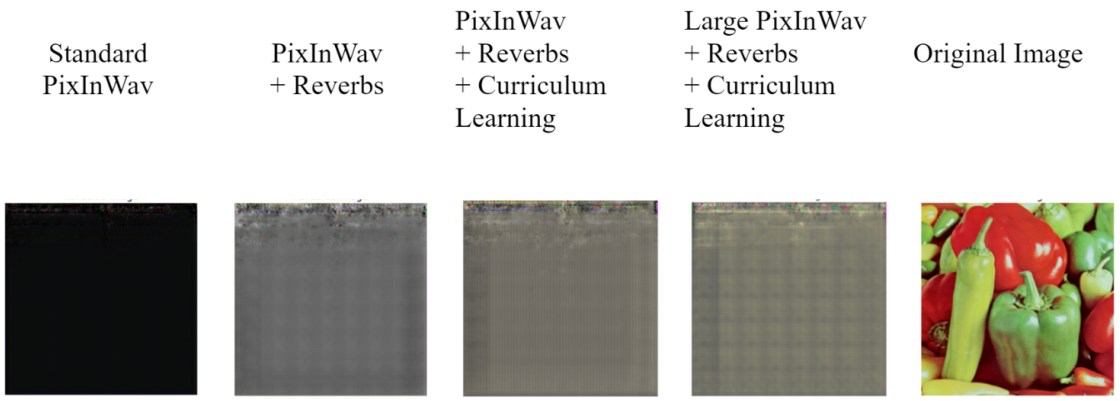

Fig 17: Perceptual results of the maximum reverberation case for the different models

Chapter 6 – Ethics

In the PixInWav project, our main goal is to transmit images inside of audio, and specifically, in this thesis, we focused on the over-the-air transmission case. Although the utilisation of this system is strictly limited to this simple problem, we believe that the Steganography is a very promising field that could be very useful in the near future to assure individual privacy in a lot of areas and that the investigation of fields like multimodal steganography is stepping in the right direction. We consider that we could complement future works, such as hiding sensible data in signals in which an attacker would not think they could be and that if he knew where they were hidden, he could not retrieve them.

In our PixInWav system, we use deep neural networks, which are known to have a high energy expenditure. In our specific case, this is not far from different, the PixInWav network consists of over 950.000 learnable parameters, which used to take days of CPU and GPU consumption to train. Despite that in Catalonia, which is the location of the servers where train and execute the models, only 25% of the energy produced is from non-renewable sources (according to the Institut Català d'Energia), given the context of the global energy crisis we do not want to use more energy than what is strictly necessary for our investigations.

To address this, we performed modifications to the code that reduce the energy consumption like reducing the number of validation steps, we optimised the code as much as possible and we always launched test experiments before launching long experiments, to avoid spending double the necessary energy. About the change in validation steps, by reducing the frequency of validation, we reduced 90% of the time of execution of our experiments.

Lastly, the fact that Steganography has been widely used to hide messages in wars (especially in World War II) is something to care about. Our investigation is fully focused on being a technological advance in the Steganography field, and we completely reject the use of this technology in the Belic field.

Chapter 7 - Conclusions and Further Research

In this thesis, we extended the PixInWav architecture to approach the over-the-air transmission problem, where the container audio was reproduced by a speaker and recorded with a microphone before passing through the decoder network. To do so, we decided to add reverberation during the training using a Curriculum Learning schedule, progressively increasing the difficulty of the reverbs fed to the network.

We performed a series of experiments, where we found that:

- Using reverberation during training improves the performance of the system both in the original PixInWav task, but also in an artificially reverberated PixInWav dataset (using reverbs that the network has never seen before).
- Using Curriculum Learning improved the performance of the network, obtaining substantially better results in terms of the revealed image in both the original and the reverberated validation tasks at the expense of a slight degradation of the container audio quality (an unnoticeable difference when hearing the container audios).
- Extending the size of the network by increasing the number of channels didn't improve the performance of the system.

Given the results, we conclude that the use of Curriculum Learning to train PixInWav using reverberation improved significantly the performance of the system in a reverb-based modelization of Over-The-Air transmission task, but as I state in the following subsection, the models proposed in this thesis should be tested in a real scenario with a speaker and recorder to assess their performances.

7.1. Further Research:

The first and main point that proceeds my investigation on the Over-The-Air transmission problem, would be to test the systems implemented and trained in this thesis in real environments, to assess the performance of these models. We expect them to obtain a better

performance than the PixInWav models trained without reverberation, but we want to make sure that we don't face a generalisation problem when facing a real case scenario.

Furthermore, regarding some of the approaches to the problem, we considered two clear lines of research:

- One pathway would be to continue in the end-to-end approach but add more kinds of reverberation, noises, and artefacts to the signals. Also, it could be interesting to find strategies to model the effect of the speaker and the microphone on the signal.

- The other pathway would be to estimate the environmental impulse response of the signal to then use it to de-reverb the container signal. In this approach, we could estimate this impulsional response by sending pilot signals, using an image of the environment, or even with the container audio itself.

Lastly, it could be interesting to increase the size of the network by adding more layers, since adding more channels did not improve the performance of the system.

References

[1] Verma, Vikas, et al. "An enhanced Least Significant Bit steganography method using midpoint circle approach", ICCSP 2014, https://ieeexplore.ieee.org/document/6949808

[2] Tancik, Matthew, et al. 'StegaStamp: Invisible Hyperlinks in Physical Photographs'. ArXiv:1904.05343 [Cs], Mar. 2020. arXiv.org, http://arxiv.org/abs/1904.05343.

[3] Zhu, Jiren, et al. 'HiDDeN: Hiding Data With Deep Networks'. ECCV 2018, https://openaccess.thecvf.com/content_ECCV_2018/papers/Jiren_Zhu_HiDDeN_Hiding_Data_ECCV_2018_paper.pdf

[4] Kreuk, Felix, et al. 'Hide and Speak: Towards Deep Neural Networks for Speech Steganography'. Interspeech 2020, https://www.isca-speech.org/archive_v0/Interspeech_2020/pdfs/2380.pdf

[5] Geleta, Margarita, et al. 'PixInWav: Residual Steganography for Hiding Pixels in Audio', ICASSP 2022, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9746191

[6] Traer, McDermott et al. 'Statistics of natural reverberation enable perceptual separation of sound and space', PNAS 2016, https://mcdermottlab.mit.edu/Reverb/IR_Survey.html

[7] Shivakumar, Prashanth & Georgiou, Panayiotis. Perception Optimised Deep Denoising AutoEncoders for Speech Enhancement. Interspeech 2016, https://www.isca-speech.org/archive_v0/Interspeech_2016/pdfs/1284.PDF

[8] Grozdić, Đorđe & Jovicic, Slobodan. 'Whispered Speech Recognition Using Deep Denoising Autoencoder and Inverse Filtering'. IEEE/ACM Transactions on Audio, Speech, and Language Processing 2017, https://ieeexplore.ieee.org/document/8114355

[9] Hegde, Sindhu , et al. 'Visual Speech Enhancement Without A Real Visual Stream'. ArXiv:2012.10852 [Cs, Eess], Dec. 2020. arXiv.org, http://arxiv.org/abs/2012.10852.

[10] Steinmetz, Christian J., et al. 'Filtered Noise Shaping for Time Domain Room Impulse Response Estimation From Reverberant Speech'. WASPAA, 2021, https://facebookresearch.github.io/FiNS/

[11] Zhong-Hua Fu et. al. 'GPU-based image method for room impulse response calculation'. Multimed Tools Appl 2016, https://doi.org/10.1007/s11042-015-2943-4

[12] Singh, Nikhil, et al. 'Image2Reverb: Cross-Modal Reverb Impulse Response Synthesis', ICCV 2021, https://openaccess.thecvf.com/content/ICCV2021/papers/Singh_Image2Reverb_Cross-Modal_Reverb_Impulse_Response_Synthesis_ICCV_2021_paper.pdf

[13] Perez, Luis & Wang, Jason. (2017). 'The Effectiveness of Data Augmentation in Image Classification using Deep Learning'. http://arxiv.org/abs/1712.04621.

[14] Bengio, Y. & Louradour, Jérôme & Collobert, Ronan & Weston, Jason. (2009). 'Curriculum learning. Journal of the American Podiatry Association'. https://www.researchgate.net/publication/221344862_Curriculum_learning

[15] Soviany, Petru, et al. 'Curriculum Learning: A Survey'. IJCV 2021, http://arxiv.org/abs/2101.10382.

[16] Braun, Stefan, et al. 'A Curriculum Learning Method for Improved Noise Robustness in Automatic Speech Recognition'. EUSIPCO 2016, https://www.eurasip.org/Proceedings/Eusipco/Eusipco2017/papers/1570341635.pdf

[17] Amodei, Dario, et al. 'Deep Speech 2: End-to-End Speech Recognition in English and Mandarin'. https://proceedings.mlr.press/v48/amodei16.pdf

[18] Ronneberger, Olaf, et al. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. MICCAI 2015, https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

[19] Fonseca, Eduardo et. al. 'Learning Sound Event Classifiers from Web Audio with Noisy Labels', ICASSP 2019, https://repositori.upf.edu/bitstream/handle/10230/42576/fonseca_icassp19_lear.pdf?sequence=1&isAllowed=y

[20] Russakovsky, Olga & Deng, Jia et. al. 'ImageNet Large Scale Visual Recognition Challenge'. IJCV, 2015, https://link.springer.com/article/10.1007/s11263-015-0816-y

[21] Wang, Zhou et. al. 'Image quality assessment: from error visibility to structural similarity,' in IEEE Transactions on Image Processing 2004, https://ieeexplore.ieee.org/document/1284395

Appendix 1: Complementary Files

This document is located in the directory PixInWav. Note that inside of this directory, there is also:

- The ‘audios’ directory, with container audios obtained by the models as well as the original audio.
- The ‘src’ directory, containing the source code used for this project.