ASMR Is All You Need

Víctor Adell*
UPC Barcelona
victor.adell

Jordi Aguilar*
UPC Barcelona
jordi.aguilar.larruy

Pau Autrand*
UPC Barcelona
pau.autrand

Miquel Escobar*
UPC Barcelona
miquel.escobar

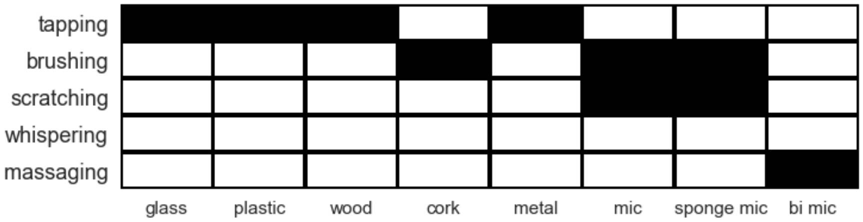
Abstract

In the past few years the popularity of the autonomous sensory meridian response (ASMR) concept has risen exponentially, being the video format the most common source for its stimuli. These are videos made by content creators who generate trigger sounds by using multiple materials and techniques. Therefore, in this study we propose variations of the WaveRNN and WaveNet models designed to generate these trigger sounds from scratch. We observe that the baseline architecture of these models outperform the alternative conditioned models in terms of the quality of the generated audios.

1 Introduction

According to Wikipedia [1], an autonomous sensory meridian response (ASMR) is a tingling sensation that typically begins on the scalp and moves down the back of the neck and upper spine. Thus, ASMR audiovisual content consists of a collection of sounds with the intent of producing this tingling sensation on the listener. The popularity of this content is enormous and growing: in fact, there are over 13 million published videos on YouTube [2].

The content creators often produce the sounds by applying various techniques to different materials: for instance, they repeatedly *tap* a surface made of glass, wood or plastic with their fingers or *brush* the microphone, among others. The most common techniques and materials (the combination of both is known as *trigger*) are *tapping*, *brushing*, *scratching*, *whispering* and *ear massaging*, as shown in figure 1.



	glass	plastic	wood	cork	metal	mic	sponge mic	bi mic
tapping	■	■	■	□	■	□	□	□
brushing	□	□	□	■	□	■	■	□
scratching	□	□	□	□	□	■	■	□
whispering	□	□	□	□	□	□	□	□
massaging	□	□	□	□	□	□	□	■

Figure 1: Most common combinations of techniques and materials, marked in black.

The purpose of this study is to generate audio samples that resemble those pertaining to these popular videos in social media. In order to do so, we have trained the `WaveRNN` and `WaveNet` models using data from YouTube videos, as thoroughly explained in **section 3.1**.

The results of our work can be found in this web page. The generated samples are displayed along with the original ones, in order to provide the user with a reference for comparison purposes.

The source code of the project can be found on this repository.

*Equal contribution. Listing order is alphabetical. The domain of all email addresses is `est.fib.upc.edu`.

2 Approach

The characteristics of ASMR sounds depend greatly on the specific type of ASMR. However, we use the same model to generate audio samples for all ASMR triggers. To do so, two very different architectures have been tested: the `WaveNet` and the `WaveRNN` deep neural networks. Sequential models achieve state-of-the-art results in audio [3], visual and textual domains with respect to both estimating the data distribution and generating high-quality samples, being the `WaveNet` the father of them all. Nonetheless, efficient sampling for this class of models has remained an elusive problem. This lead to the design of other architectures such as the `WaveRNN`, which matches the quality of the `WaveNet` but is computationally more efficient.

That being said, `WaveNet` is a much more studied model [4], which implies that features such as global and local conditioning do not need to be designed from scratch. Hence, although the computational resources are very limited for this project, results with a conditioned `WaveNet` appear to be better than the ones with the conditioned `WaveRNN`. Nevertheless, the baseline model for the `WaveRNN` seems to perform better than `WaveNet`'s.

In the following sections both architectures are thoroughly discussed, as well as our modifications and adaptations, which are also described.

2.1 WaveRNN

2.1.1 Baseline model

The `WaveRNN` model is a deep architecture for speech synthesis, consisting in a single-layer RNN with a dual softmax layer that is designed to efficiently predict 16-bit raw audio samples. Its main benefits are the low number of parameters with respect to the `WaveNet` model while achieving similar quality, and the possibility of generating real time audio on a broad set of computing platforms, including mobile CPUs. In summary, the `WaveRNN` model seeks an architecture that provides an equally expressive and non-linear transformation of the context, but also requires a small number of operations at each step. Figure 2 illustrates its architecture.

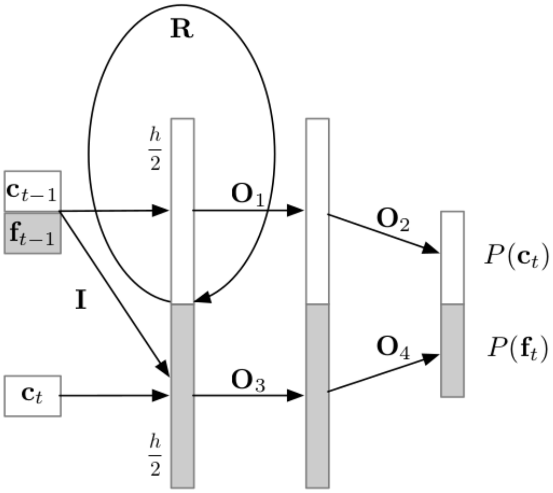


Figure 2: `WaveRNN` architecture [3].

One of the peculiarities of the `WaveRNN` is distinguishing between the coarse (high 8-bits) of the sample and the fine (low 8-bits) of the sample, both in the input and the output of the network. The multiplication by **R** is computed simultaneously for both the coarse and fine bits, and then the output of the gates is evaluated for the coarse bits only while $\mathbf{c_t}$ is sampled. Once $\mathbf{c_t}$ has been sampled from $P(\mathbf{c_t})$, the gates are evaluated for the fine bits and $\mathbf{f_t}$ is sampled. The resulting *dual soft-max* layer allows an efficient prediction of 16-bit samples using only two small output spaces (2^8 values each, a combined total of 2^9), instead of a single large output space (with 2^{16} values). The overall sequential computation of the `WaveRNN` is as follows (biases are omitted for brevity purposes):

$$
\begin{aligned}
&\mathbf{x}_t = [\mathbf{c}_{t-1}, \mathbf{f}_{t-1}, \mathbf{c}_t] \\
&\mathbf{u}_t = \sigma\left(\mathbf{R}_u \mathbf{h}_{t-1} + \mathbf{I}_u^\star \mathbf{x}_t\right) \\
&\mathbf{r}_t = \sigma\left(\mathbf{R}_r \mathbf{h}_{t-1} + \mathbf{I}_r^\star \mathbf{x}_t\right) \\
&\mathbf{e}_t = \tau\left(\mathbf{r}_t \circ \left(\mathbf{R}_e \mathbf{h}_{t-1}\right) + \mathbf{I}_e^\star \mathbf{x}_t\right) \\
&\mathbf{h}_t = \mathbf{u}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \circ \mathbf{e}_t \\
&\mathbf{y}_c, \mathbf{y}_f = \operatorname{split}\left(\mathbf{h}_t\right) \\
&P\left(\mathbf{c}_t\right) = \operatorname{softmax}\left(\mathbf{O}_2 \operatorname{relu}\left(\mathbf{O}_1 \mathbf{y}_c\right)\right) \\
&P\left(\mathbf{f}_t\right) = \operatorname{softmax}\left(\mathbf{O}_4 \operatorname{relu}\left(\mathbf{O}_3 \mathbf{y}_f\right)\right)
\end{aligned}
\tag{1}
$$

Where $\star$ indicates a masked matrix whereby the last coarse input $\mathbf{c}_t$ is only connected to the fine part of the states $\mathbf{u}_t$, $\mathbf{r}_t$, $\mathbf{e}_t$ and $\mathbf{h}_t$. Thus, it only affects the fine output $\mathbf{y}_f$. The coarse and fine parts are encoded as values in $[0, 255]$ and scaled to the interval $[-1, 1]$. The matrix $\mathbf{R}$ formed from the matrices $\mathbf{R_u}$, $\mathbf{R_r}$, while $\mathbf{R_e}$ is computed as a single matrix-vector product to produce the contributions to all three gates $\mathbf{u_t}$, $\mathbf{r_t}$ and $\mathbf{e_t}$ (a variant of the GRU [5]).

2.1.2 Conditioned model

So far, we have seen the common implementation of the `WaveRNN` as proposed in the original paper [3]. However, we would prefer to condition the `WaveRNN` in order to be able to force a specific kind of ASMR sound to the network, similar to the task of multi-speaker speech generation. One of the challenges that we have faced is the lack of clear documentation and information on this subject. The approach that we suggest is passing the conditioning features through a dense layer, and then biasing the gates and concatenating the embeddings of the features to the hidden layer.

The modified equations to bias the gates according to a set of features $\mathbf{h}$ are:

$$
\begin{aligned}
&\mathbf{x}_t = [\mathbf{c}_{t-1}, \mathbf{f}_{t-1}, \mathbf{c}_t] \\
&\mathbf{u}_t = \sigma\left(\mathbf{R}_u \mathbf{h}_{t-1} + \mathbf{I}_u^\star \mathbf{x}_t + \mathbf{V}_u \mathbf{h}\right) \\
&\mathbf{r}_t = \sigma\left(\mathbf{R}_r \mathbf{h}_{t-1} + \mathbf{I}_r^\star \mathbf{x}_t + \mathbf{V}_r \mathbf{h}\right) \\
&\mathbf{e}_t = \tau\left(\mathbf{r}_t \circ \left(\mathbf{R}_e \mathbf{h}_{t-1}\right) + \mathbf{I}_e^\star \mathbf{x}_t + \mathbf{V}_e \mathbf{h}\right) \\
&\mathbf{h}_t = \mathbf{u}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \circ \mathbf{e}_t \\
&\mathbf{y}_c, \mathbf{y}_f = \operatorname{split}\left(\mathbf{h}_t\right) \\
&P\left(\mathbf{c}_t\right) = \operatorname{softmax}\left(\mathbf{O}_2 \operatorname{relu}\left(\mathbf{O}_1 \mathbf{y}_c\right)\right) \\
&P\left(\mathbf{f}_t\right) = \operatorname{softmax}\left(\mathbf{O}_4 \operatorname{relu}\left(\mathbf{O}_3 \mathbf{y}_f\right)\right)
\end{aligned}
\tag{2}
$$

The modified equations to concatenate the embeddings $\mathbf{b}$ to the hidden layer are:

$$
\begin{aligned}
&\mathbf{x}_t = [\mathbf{c}_{t-1}, \mathbf{f}_{t-1}, \mathbf{c}_t] \\
&\mathbf{h}_{t-1}^b = [\mathbf{h}_{t-1}, \mathbf{b}] \\
&\mathbf{u}_t = \sigma\left(\mathbf{R}_u \mathbf{h}_{t-1}^b + \mathbf{I}_u^\star \mathbf{x}_t + \mathbf{V}_u \mathbf{h}\right) \\
&\mathbf{r}_t = \sigma\left(\mathbf{R}_r \mathbf{h}_{t-1}^b + \mathbf{I}_r^\star \mathbf{x}_t + \mathbf{V}_r \mathbf{h}\right) \\
&\mathbf{e}_t = \tau\left(\mathbf{r}_t \circ \left(\mathbf{R}_e \mathbf{h}_{t-1}^b\right) + \mathbf{I}_e^\star \mathbf{x}_t + \mathbf{V}_e \mathbf{h}\right) \\
&\mathbf{h}_t = \mathbf{u}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{u}_t) \circ \mathbf{e}_t \\
&\mathbf{y}_c, \mathbf{y}_f = \operatorname{split}\left(\mathbf{h}_t\right) \\
&P\left(\mathbf{c}_t\right) = \operatorname{softmax}\left(\mathbf{O}_2 \operatorname{relu}\left(\mathbf{O}_1 \mathbf{y}_c\right)\right) \\
&P\left(\mathbf{f}_t\right) = \operatorname{softmax}\left(\mathbf{O}_4 \operatorname{relu}\left(\mathbf{O}_3 \mathbf{y}_f\right)\right)
\end{aligned}
\tag{3}
$$

2.2 WaveNet

2.2.1 Baseline model

The `WaveNet` architecture [4] was introduced in 2016 and it entailed a change of paradigm in speech synthesis given that it is a generative model that directly operates at the lowest possible level of raw audio waveforms. Therefore, in this section we will outline the main components of this model.

The long-term dependence among sequential waveform samples $\mathbf{x} = \{x_1, \dots, x_T\}$ is modelled as a product of conditional probabilities:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p\left(x_t | x_1, \dots, x_{t-1}\right) \tag{4}$$

The conditional probability distribution is modelled by a stack of causal convolutional layers such that the prediction at time step t does not depend on any of the future time steps. Moreover, the convolutions are dilated in such a way that the filters are applied over an area larger than its length by skipping input values with a certain step. As shown in Figure 3, this allows the network to have very large receptive fields with just a few layers.

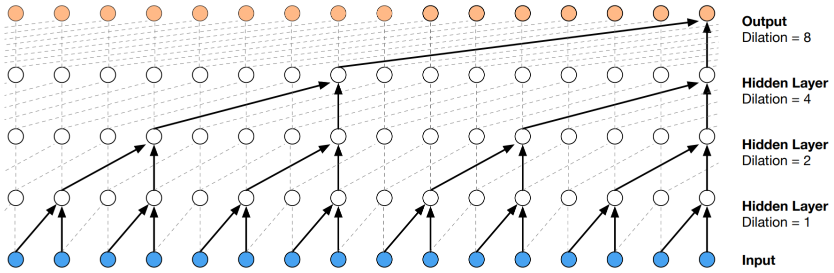


Figure 3: Visualization of a stack of causal convolutional layers [4].

At the end of the convolutional structure, the `WaveNet` architecture transforms the waveform generation into a classification problem. By encoding the signal into 8 bits using the μ-law, we have a non-linear quantization that produces a significantly better reconstruction than a simple linear quantization scheme [4]. In addition, a gated structure is applied to enhance the modelling capability, which is formulated as

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \odot \sigma\left(W_{g,k} * \mathbf{x}\right), \tag{5}$$

where k is the layer index, f and g denote filter and gate, and W is a learnable convolution filter. As mentioned in the original paper, residual and skip connections are used throughout the network to speed up convergence and enable training of deeper models. To sum up, previous samples pass through a causal layer and several residual blocks which contain a dilated convolutional layer, a gated activation and residual and skip connections. Finally, the summation of all skip connections passes to two 1×1 convolutions and one softmax layer that outputs the predicted distribution of the current sample. The full architecture for the `WaveNet` model is displayed in Figure 4.

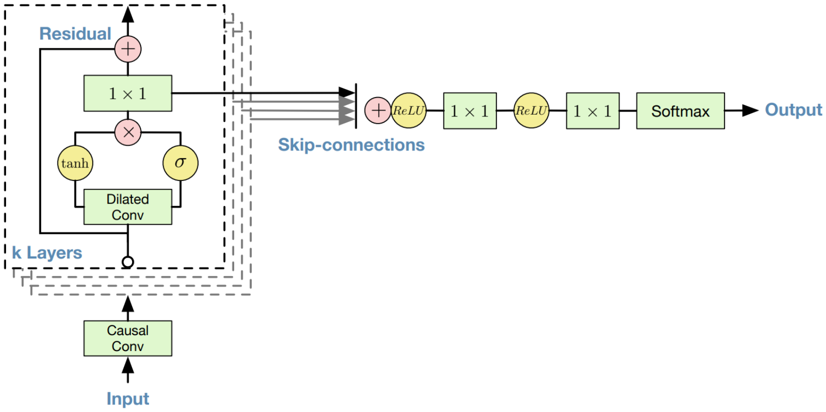


Figure 4: Overview of the residual block and the entire architecture [4].

2.2.2 Conditioned model

In order to guide the WaveNet's generation towards required characteristics, the authors suggested conditioning the model on other input variables, either globally or locally.

In the context of ASMR synthesis, we are interested in conditioning the overall generation to the different triggers. Hence, global conditioning in the WaveNet is characterised by a single latent representation $\mathbf{h}$ that influences the output distribution across all time steps. The activation function from equation 5 now becomes

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x} + V_{f,k}^{T}\mathbf{h}\right) \odot \sigma\left(W_{g,k} * \mathbf{x} + V_{g,k}^{T}\mathbf{h}\right), \tag{6}$$

where $V_{*,k}$ is a learnable linear projection and the vector $V_{*,k}^{T}\mathbf{h}$ is broadcast over the time dimension.

In the multi-speaker speech generation experiment mentioned on the WaveNet paper [4], the conditioning was applied by feeding the speaker identifier to the model as a one-hot vector. In the conditioned ASMR generation, however, we have directly implemented the $V_{*,k}^{T}\mathbf{h}$ vectors as an embedding module.

3 Experiments

3.1 Data

In this section we walk through how we obtain and transform the data to the format required by our models. Please refer to the notebooks in ./notebooks/preparation/, as well as the utils module in ./notebooks/preparation/ and the youtube module in ./src/ in the repository for more details.

3.1.1 Data extraction

All the data used for this project is extracted from YouTube videos. The amount of ASMR videos in YouTube is vast, and their content can be accessed through its API. Even so, it is clearly an unstructured and, more importantly, unlabeled data source.

In order to crop and label the specific sounds reproduced in a video, which is a collection of different triggers, we extract this information from either the description of the video or the comments, where we usually find the timestamps of each trigger that appears in the video.

After this transformation, the cropped and labeled audios are stored in ./data/processed/, where they are subdivided by trigger type.

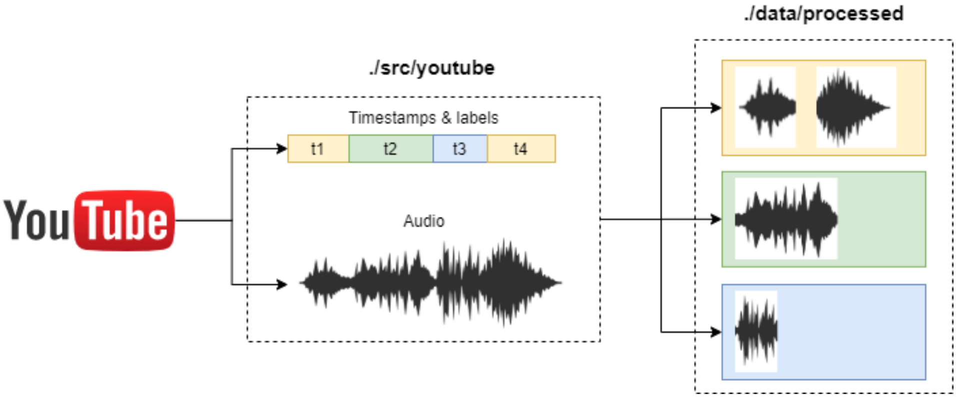


Figure 5: Schema of the data ETL process.

3.1.2 Exploratory analysis

The different ASMR triggers have very distinct wave shapes. Moreover, there exist some key differences between ASMR audios and voice signals, the latter being the most common input for the models used in this study. Therefore, in order to better understand if the same models can be applied to all of them or if it has to be adapted, we have made an exploratory analysis of the data.

Figure 6 shows the plot of *tapping*, *brushing* and *whispering* ASMR triggers, as well as a human voice signal.

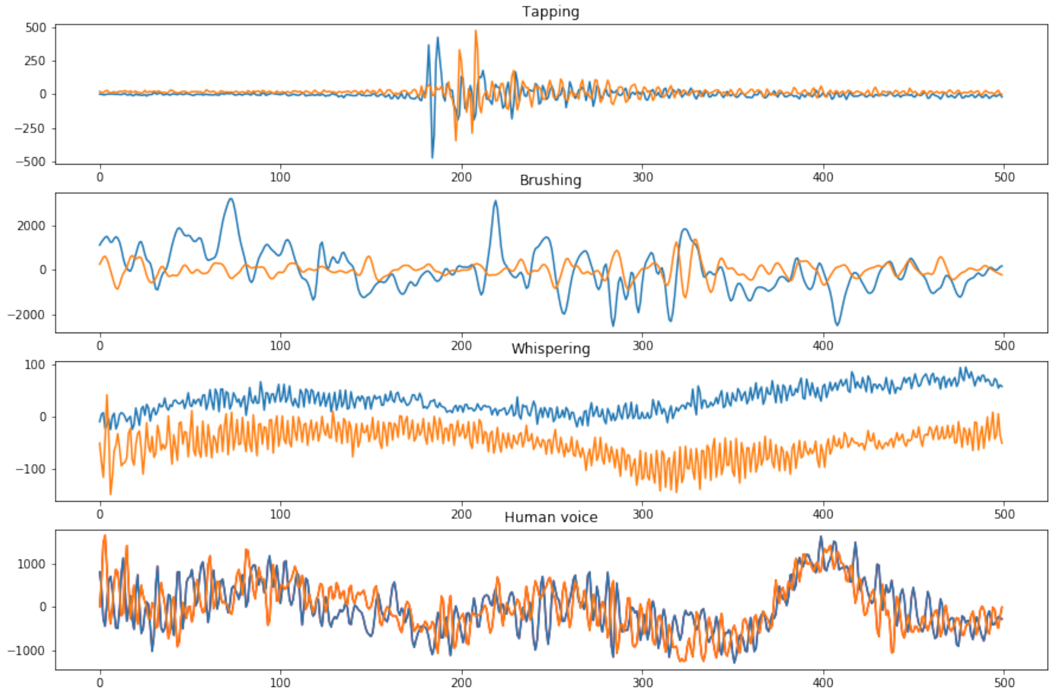


Figure 6: *Tapping*, *brushing*, *whispering* and *speaking* raw stereo signals, from top to bottom.

Firstly, it is important to mention that there are two signals in each of the axes. That is because they are extracted in stereo format, and then converted to mono format. It can be observed that the *tapping* audio is constant most of the time, but with big sporadic peaks. It has therefore high frequencies. On the other hand, in the *brushing* audio we can find lower frequencies. Finally, the *whispering* audio is the most similar to the voice signal, but much softer and with a lower variance.

To have a different perspective, it is interesting to also look at the signals not only on the time domain but also in the frequency domain, which is shown in figure 7, by means of an STFT plot.

We observe what we could expect based on figure 6. The *tapping* spectogram is of low energy in most time steps, but there exist sporadic peaks that correspond to a wide range of frequencies. In the *brushing* audio spectogram, we see that the frequency distribution is quite constant over time, without a distinguishable pitch. The *whispering* spectogram contains lower frequencies in comparision, but the pitch and formants start being more distinguishable (as we are treating with human voice), and the same, but more marked, can be said about the normal human voice histogram.

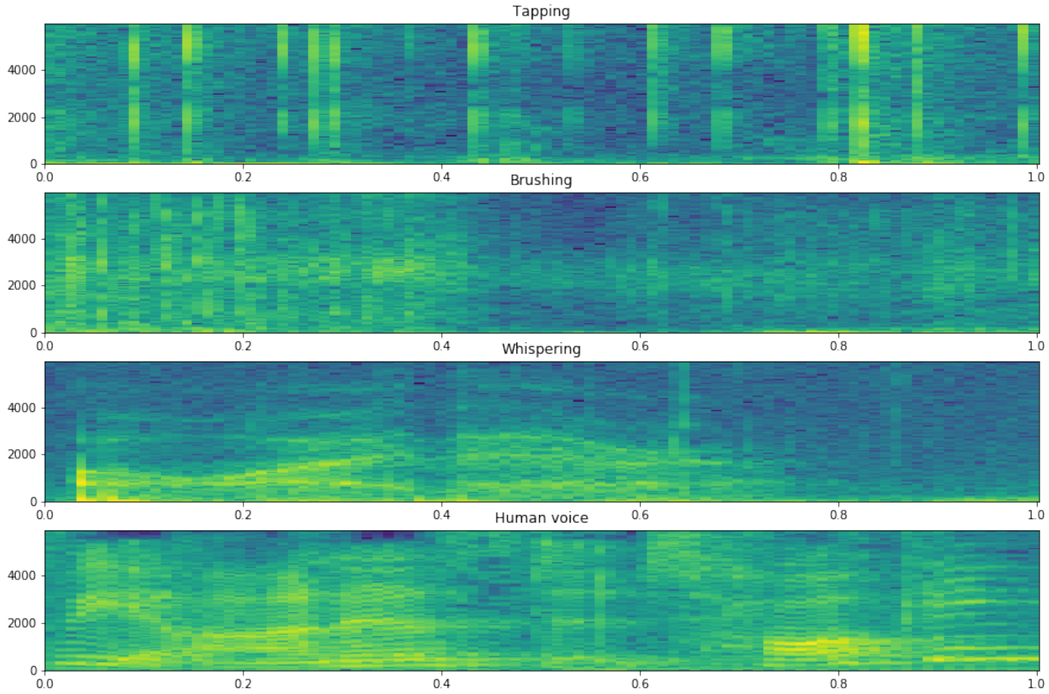


Figure 7: *Tapping*, *brushing*, *whispering* and *speaking* STFT, from top to bottom.

3.2 Hyperparameters

The fact that popular and therefore very studied models are being used implies that the typical hyperparameter optimisation has already been done. However, ASMR audio differs quite a bit from speech or even music audio signals, and consequently some hyperparameters were slightly modified in the `WaveRNN` model.

The first thing that was changed was the sample rate. The extracted YouTube audios had a sample rate of 40kHz. Nonetheless, the code used was designed for an audio of 22.5kHz, and had, therefore, a window length adequate for this sample rate. First, the model was trained at 40kHz, but the results were much worse than the results after transforming the sample rate to 22.5kHz. The reason is that if the window length is kept constant, the higher the sample rate the lower the time steps the RNN is keeping in memory, so the results get much worse. The opposite was also tested, i.e. keeping the sample rate at 40kHz while increasing the window length proportionally. However, the model was slower and the results were not better, probably because the RNN was not able to retain useful information on very old data or this information was simply not relevant enough to improve the model.

Secondly, the parameters of the loss function were tested. The loss function is simply the sum of the loss of the coarse part and the fine part, but it was interesting to try to change the ponderation of loss to the total one. However, the results for the best performing ponderations did not seem to improve the results significantly enough to conclude that they were actually improving the model, and some of them even gave worse results. Thus, he initial ponderation was kept.

Regarding the implementation of the `WaveNet` architecture, Table 1 below summarizes the configuration of the most relevant hyperparameters that have been used.

3.3 Qualitative results

The conditioned `WaveRNN` that we have introduced in this report has yielded to poor results when trying to generate sounds with different kinds of objects, even with the same type of ASMR sound. The neural network is able to generate sounds that imitate the rhythm and the intensity of the desired sound, but it doesn't differentiate as good as we would expect between a specific ASMR sound

Table 1: Relevant hyperparameters for the `WaveNet` implementation.

Hyperparameter	**Value**
Number of layers	10
Stacks of conv. layers	5
Channels for input data	256
Residual channels	512
Sample rate	16 kHz
Time steps trained at once	32,000

applied to different materials. The reason why we are experiencing this issue may be the lack of robustness of the conditioned approach proposed in this report and the big difference between the sound produced by different objects, covering solid, hollow, metallic or glassy sounds, among others.

The conditioned `WaveNet`'s quality of the results is lower than in the baseline models trained on a single type of ASMR technique and material. In fact, the generated waveforms are noisier and the sound patterns are not clearly perceived.

In order to gain some insights, we have focused on the latent representations $\mathbf{h}$ that the network has learned for each one of the ASMR types. They consist on the glass, plastic, wood and metal tapping triggers as well as brushing triggers on a cork surface, a microphone and a sponge microphone.

More concretely, we have extracted their embeddings and computed the pairwise Euclidean distances between them. We have repeated this process at two different training stages, both the 100th and 10,000th epoch. In order to visualize the evolution they had, we have used the Multidimensional Scaling (MDS) technique, which maps a distance matrix into a Cartesian space defined to be two-dimensional. The results are shown in Figure 8 below.

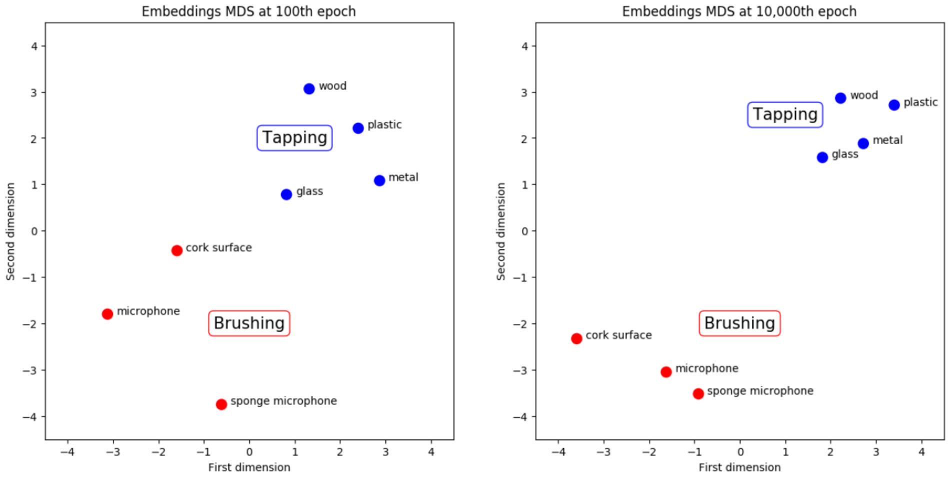


Figure 8: Multidimensional scaling of global embeddings at different epochs.

The fact that conditioning the network yields two distant clusters of embeddings corresponding to the *tapping* and *brushing* triggers leads us to think that training a distinct `WaveNet` for each trigger is a better option to generate higher quality outputs.

In fact, in multi-speaker speech generation the addition of different speakers results in better validation set performance compared to training solely on a single speaker [4]. This suggests that `WaveNet`'s internal representation is shared among multiple speakers. Nevertheless, in the context of ASMR synthesis we have empirically checked that the different nature of triggers, such as tapping or brushing, is not suitable for training a unified conditioned model.

It is interesting to check the difference between one of the original ASMR audios and one generated by one of our models. Below we compare an original glass tapping sample with a generated one, both in the time and frequency domain (see figure 9). It can be observed that the main problem is the presence of noise, but the shape and behaviour of the wave is quite good. The cause of the noise is probably the fact that we are trying to generate a sound composed mainly for high frequencies, which is precisely the main characteristic of noise.

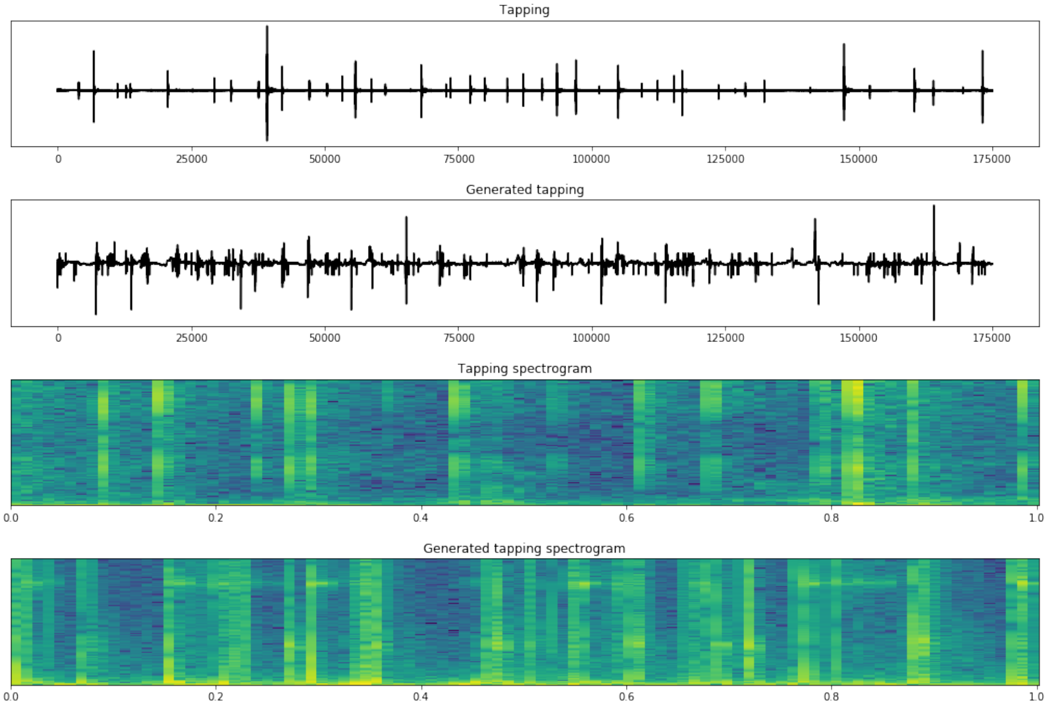


Figure 9: Original glass tapping vs. generated glass tapping.

4 Next steps

After all the presented work and the achieved results, we still believe that there exist some tweaks that we can perform in order to improve the quality of the generated ASMR audios.

One the one hand, we should focus on the proper implementation of the conditioned network so that we can create any kind of ASMR sound from a single neural network. This is, in fact, our ultimate goal. To do so, it is essential to have enough time to train our models. The results shown are generated from neural networks that have been trained during limited time due to finite resources.

On the other hand we must point out that our results have a remarkable difference with respect the original audios, that is the stereo display. Both the output and the input of the common speech synthesis architectures do not take into account the audio's format (e.g. stereo instead of mono), and are designed to work with 1-D signals. However, ASMR audios are conceived to be listened using headphones, taking advantage of the oscillation of the sound between the right and left channel to increase the user's enjoyment listening to the audio. There is certainly a niche with the task of generating stereo sound instead of mono, where innovative ideas involving deep neural networks can play an important role.

5 Conclusions

In this project we have implemented a model to generate ASMR audios which is, very probably, the very first one on this subject. For this reason, we could say that we have had two more added challenges than any project in which there exists previous research.

Firstly, there does not exist any kind of dataset whatsoever related to our purposes. This adds the difficulty of having to generate our own dataset, which obviously must be of the maximum possible quality in order to optimize the results.

Secondly, there is hardly any information on the project's topic, let alone scientific research. This has translated into having to collect information from sources related to ASMR generation, such as music generation or speech synthesis. Thus, we often have had to rely on intuition or apply certain adaptations to better fit our data and objectives, not always obtaining top results.

To wrap up, we can say that even though these added work might have impacted negatively the final results, it has been challenging for us and we have learned the process of building a working model starting literally from scratch. It has also highlighted the critical importance of a good dataset, which is as important or even more than the model itself. Thus, we consider this project a success.

Finally we would like to mention that we have enjoyed a lot developing this study, reason for which we would like to continue working on it in order to obtain better results and, as a final goal, publish this paper.

Acknowledgments

We would like to thank José Adrián Rodríguez Fonollosa for the entire Spoken and Written Language Processing course, which has provided us with a great amount of knowledge and tools. It has been a really interesting course, in which we have learnt numerous concepts and new manners of approaching and attacking problems. In general, we can confidently claim that our skill sets have broadened significantly, for which we consider the course a total success from our part. We would also like to specifically acknowledge the work of both Aaron van den Oord and Nal Kalchbrenner, as this project is based in their research.

References

[1] The Free Encyclopedia Wikipedia. *ASMR*. 2020. URL: https://en.wikipedia.org/wiki/ASMR.

[2] University of Sheffield. *Brain tingles: First study of its kind reveals physiological benefits of ASMR*. 2018. URL: https://www.sciencedaily.com/releases/2018/06/180621101334.htm.

[3] Nal Kalchbrenner et al. *Efficient Neural Audio Synthesis*. 2018. arXiv: 1802.08435 [cs.SD].

[4] Aaron van den Oord et al. *WaveNet: A Generative Model for Raw Audio*. 2016. arXiv: 1609.03499 [cs.SD].

[5] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. arXiv: 1412.3555 [cs.NE].